MolQuest Help

MolQuest

Version 2.4

Programs Help

CONTENTS

Alignments

ESTMap

Program for mapping a whole set of mRNAs/ESTs to a chromosome sequence. For example, 11,000 sequences of full mRNAs from NCBI reference set were mapped to 52-MB unmasked Y chromosome fragment in about 18-25 min, depending on computer memory size. ESTMap takes into account statistical features of splice sites for more accurate mapping.

ESTMap is part of FGENESH++C genome annotation pipeline, where it maps RefSeq sequences to a query genome at very early stages of annotation.

```
L:4000001       Sequence  chr7  [cut:73000000  77000000]  vs  C:\Documents  and
Settings\My Documents\MolQuestWorkSpace\example_data\EstMap\seq.fa
[DD] Sequence:        1(       1), S:        36.26, L:       457 AA628013
nq61d05.s1 NCI_CGAP_Co9 Homo sapiens cDNA clone IMAGE:1148361 3', mRN
Summ of block lengths: 457, Alignment bounds:
On first  sequence: start   2214596, end   2215412, length 817
On second sequence: start         1, end       457, length 457
Block of alignment: 4
     1 E:  2214596     234 [ct CT] P:    2214596           1 L:      234, G:
99.57, W:   2305, S:26.2324
     2 E:  2214966      69 [AC CT] P:    2214966         235 L:       69, G:
100.00, W:    690, S:14.1834
     3 E:  2215144      65 [AC CT] P:    2215144         304 L:       65, G:
100.00, W:    650, S:13.7542
     4 E:  2215324      89 [AC aa] P:    2215324         369 L:       89, G:
97.75, W:    820, S:15.6754
        1 gagccaagattgtgc(..)acgctcaggccacct?[CTGGGCCTCTCTTTATTGAGGGCA
          ...............(..)...............  ||||||||||||||||||||||||
        1 ---------------(..)---------------  CTGGGCCTCTCTTTATTGAGGGCA

  2214620 CTGGGCCCAGGTCTTCCTTCAGGGCCCACAGCGCCCATAAAACCCAAGGGAGAATAGAAG
          ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
       25 CTGGGCCCAGGTCTTCCTTCAGGGCCCACAGCGCCCATAAAACCCAAGGGAGAATAGAAG

  2214680 AGACCCCCTGATACACGCACACTCGAGGGGCGCCTCCCATCCCCTCCCACAACACACAGG
          ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
       85 AGACCCCCTGATACACGCACACTCGAGGGGCGCCTCCCATCCCCTCCCACAACACACAGG

  2214740 ACAGAAGCCCCTCTGGCCGGCAGGGGAAGGCCCAGCCTCAATCCTTCTTGCTCCCGTGC
          ||||||||||||||||||||||0|||||||||||||||||||||||||||||||||||||
      145 ACAGAAGCCCCTCTGGCCGGCAAGGGAAGGCCCAGCCTCAATCCTTCTTGCTCCCGTGC

  2214800 CGCTGACTGTGAAACTTGTGGTGCACAACC]ctcagggtggtgaag(..)gggaccccgg
          |||||||||||||||||||||||||||||| ...............(..)..........
      205 CGCTGACTGTGAAACTTGTGGTGCACAACC ---------------(..)----------

  2214961 ctcac[CTGCCACTCCTTGCACTGAGGGTCCTGGGCCAGGTTGAACAACGTCAGCGCGTT
          ..... ||||||||||||||||||||||||||||||||||||||||||||||||||||||
      235 ----- CTGCCACTCCTTGCACTGAGGGTCCTGGGCCAGGTTGAACAACGTCAGCGCGTT

  2215020 AAAAAGCTGCCAGAA]ctaagcagggaggag(..)agaggcacgacttac[GTGTCCAAA
          ||||||||||||||| ...............(..)............... |||||||||
      289 AAAAAGCTGCCAGAA ---------------(..)--------------- GTGTCCAAA

  2215153 GAAAAGAAAAGGCAGCAGGAAGGTGAGGCCCCGCCACATCCAGGACTGGAAGCCCT]ctg
          |||||||||||||||||||||||||||||||||||||||||||||||||||||||| ...
      313 GAAAAGAAAAGGCAGCAGGAAGGTGAGGCCCCGCCACATCCAGGACTGGAAGCCCT ---

  2215212 cggggaggaagg(..)ccactcccgactcac[CCACAGTGAGGTCCATGGTGTGCCGCTC
```

```
        ............(..)............... ||||||||||||||||||||||||||||
    369 ------------(..)--------------- CCACAGTGAGGTCCATGGTGTGCCGCTC

2215352 GCCCAGCGCCCGCAGGCGGTAGAGGCAGCCGCTCTGGTAGTAGTACTGGAGAAACTGCAC
        ||||||||||||||||0|0|||||||||||||||||||||||||||||||||||||||||
    397 GCCCAGCGCCCGCAGGGGATAGAGGCAGCCGCTCTGGTAGTAGTACTGGAGAAACTGCAC

2215412 G]?aagcctgggccgggc(..)tacagcaaaactgga
        |  ...............(..)...............
    457 G  ---------------(..)---------------
```

Where:
1-st line is the header:

```
[DD] Sequence:         1(        1), S:         36.26, L:        457 AA628013
nq61d05.s1  NCI_CGAP_Co9  Homo  sapiens  cDNA  clone  IMAGE:1148361  3',  mRNA
sequence.
```

[DD]	Target sequence in direct chain (D), query sequence in direct chain (D). Variants: [DR] - target sequence in direct chain (D), query sequence in reverse chain (R). [RD] - target sequence in reverse chain (R), query sequence in direct chain (D). [RR] - target sequence in reverse chain (R), query sequence in reverse chain (R).
Sequence: 1(1)	Order number of sequence from a query set which is submitted to alignment. In brackets is an order number for alignment of this sequence (if it resulted in more than one alignment). Variants: 4(5) - the fifth alignment of the fourth sequence from a set
S	Score of this alignment.
L	Length of this query sequence
AA628013 nq61d05.s1 NCI_CGAP_Co9 Homo sapiens cDNA clone IMAGE:1148361 3', mRNA sequence.	Name of this query sequence

Additional information about alignment:

```
Summ of block lengths: 457, Alignment bounds:
On first  sequence: start   2214596, end   2215412, length 817
On second sequence: start         1, end       457, length 457
```

length	The length covered by alignment, in target and query sequences appropriately.

List of alignment blocks:

```
Block of alignment: 4
     1 E:  2214596     234 [ct CT] P:    2214596           1 L:      234, G:
99.57, W:   2305, S:26.2324
     2 E:  2214966      69 [AC CT] P:    2214966         235 L:       69, G:
100.00, W:    690, S:14.1834
```

Block of alignment: 4 - Number of blocks in this alignment. Each line below defines an appropriate block. Detailed description of a line from this list is shown further:

```
   1 E:  2214596    234 [ct CT] P:   2214596         1 L:     234, G:  99.57,
 W:   2305, S:26.2324
```

1	Block number.
E: 2214596 234 [ct CT]	Starting point and length of exon in the first sequence. [ct CT] - edging nucleotides of exon. Small letters - the edge is defined imprecisely. Capital letters - the edge is defined precisely.
P: 2214596 1	Positions of similarity block' start in target and query sequences appropriately.
L: 234	Length of this similarity block.
G: 99.57	Homology of this similarity block.
W: 2305	Weight of this similarity block (the arithmetic sum of symbols' similarity calculated from the given similarity matrix).
S:26.2324	Score of this similarity block.

Alignment:

```
1 gagccaagattgtgc(..)acgctcaggccacct?[CTGGGCCTCTCTTTATTGAGGGCA
  ...............(..)...............  ||||||||||||||||||||||||
1 ---------------(..)---------------  CTGGGCCTCTCTTTATTGAGGGCA
```

1 line - The target sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions. **[]** - edges of exon. **?[** - unsure edge of exon.

2 line - Separator line.

3 line - The query sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.

Parameters:

Input	
Target sequence	Place your query file with nucleotide sequences.
Query sequence(s)	Place file with one ore more nucleotide sequences.
Output	
Result	Name of the output file.
Format	Output format: **List of alignment blocks coordinates** **List of alignment blocks coordinates and blocks sequences** **Output alignment (default)** **General alignment information** **General alignment information, blocks list and alignment**
Sort blocks	Sort regions of homology for "List of alignment blocks coordinates" value of "Output format" option : **Don't sort (default)** **Incremental on Target** **Incremental on Query** **Decremental by score** **Decremental by weight** **Decremental by length**

Flank type	Flank type: **Length** - Output for given amount of symbols in flank of alignment block. **All** - unlimited flank
Position number	Print additional strings with position number for target and query strings.
Numeration Offset	Numeration Offset: **Target** - Given value will be added to taget sequence numeration on output **Query** - Given value will be added to query sequence numeration on output
Special symbols:	Special symbols: **Homology** - Output symbol as separator lines between target and query, each line separator position shows similarity between target and query positions **Gap** - Use given simbol to print output gaps **Tailing Gap** - Use given simbol to print output flanking gaps in profile output, default: '-' **Line Tearing** - String used for displaying of big gaps in alignment.
Output string	Output for given amount of symbols in each line.
Unalignment info	Produce output information for sequences where no similarity found.
Perfect only	Output perfect and near-perfect alignment.
Preprocessing	
Remove	Remove: **PolyA** - Remove polyA tail from taget sequence. It is may be useful if target sequence is mRNA or EST. **PolyT** - Remove polyT head from taget sequence. It is may be useful if target sequence is complemented mRNA or EST. **Trailing N** - Remove trailing N symbols from both ends of target sequence.
Cut Sequence	Cut Sequence: **Start** - Search in target sequence from given position **End** - Search in target sequence to given position. "0" - get to end
Apply to chain	Search in target sequence is applied to reverse chain.
Options	
Alignment accurancy	Alignment accurancy: **Weak (fast)** **Normal (slow)**
Mapping accurancy	Mapping accurancy: **Weak (fast)** **Normal (slow)**
Score method	Scoring methods for whole alignment: **No scoring the alignment (default)** **By probability of the best block in alignment** **By probability of the summary of all blocks** **Blast-like (in SD units)** **Blast-like (in probability units)**
Threshold	If alignment has score less then given value then alignment is not printed.
Target chain(s)	Search in chain(s) in target: **In direct chain only** **In reverse chain only**

	In both chains
Fine adjustment	Fine adjustment of alignment blocks ends.
Multiply variants:	**Alternate variants** - Produce given best alternate variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best alternate variants of alignments. **Non-overlapped variants** - Produce given non-overlapped variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best non-overlapped alignments. **Different variants** - Produce given different variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce different variants of alignments.
Local alignment	Produce local alignment. Split alignment to several local alignments.
Split diagonal recursively	Split diagonal recursively (if possible).
Restrictions	**Target:** **By length** - Alignment region on target sequence does not exceed given length. **By multiplier** - Alignment region on target sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on target sequence does not exceed length of query sequence plus N. **Query:** **By length** - Alignment region on query sequence does not exceed given length. **By multiplier** - Alignment region on query sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on query sequence does not exceed length of query sequence plus N.
Maximal allowed intron length	Maximal allowed intron length

GenomeMatch

Alignment of two genomes or chromosomes. Program for quick aligning of procariotic genomes, chromosomes and chromosomal contigs, genomes of mitochondria, organelles, viruses etc. Program finds relatively long similarity regions, which may contain gaps inside. Such regions may overlap each other, i.e. some nucleotides either in query or in target sequences may belong to different alignments.

Output example:

```
L:4403836         Sequence  gb|AE000516|AE000516  Mycobacterium  tuberculosis
CDC1551,          complete          genome          vs          C:\Program
Files\Softberry\MolQuest\example\data\GenomeMatch\seq2.fna
[DD] Sequence:        1(      14), S:        726.8, L:  4411529 emb|AL123456|
MTBH37RV Mycobacterium tuberculosis complete genome
Summ of block lengths: 176235, Alignment bounds:
On first  sequence: start   1266719, end   1442971, length 176253
On second sequence: start   1267228, end   1443483, length 176256
Block of alignment: 9
    1 P:   1266719   1267228 L:   10640, G:  99.98, W: 106350, S:178.608
    2 P:   1277360   1277868 L:    6697, G:  99.90, W:  66760, S:141.524
    3 P:   1284070   1284580 L:   26749, G:  99.98, W: 267317, S:283.187
```

```
4 P:   1310820   1311331 L:    2005, G: 100.00, W:  20050, S:77.5178
5 P:   1312827   1313337 L:      53, G: 100.00, W:    530, S:12.3781
6 P:   1312880   1313391 L:   52449, G:  99.96, W: 523830, S:396.44
7 P:   1365330   1365840 L:   23182, G:  99.99, W: 231720, S:263.654
8 P:   1388512   1389023 L:   20355, G:  99.99, W: 203470, S:247.058
9 P:   1408867   1409379 L:   34105, G:  99.98, W: 340857, S:319.777
1266704   1266704   1266705   1266715   1266725   1266735
      ---------------(..)tgggaccgccattgcCGGCCGTTCCACGGCCCGTATCGTC
      ...............(..)...............|||||||||||||||||||||||||
      ttgaccgatgacccc(..)tgcgcggcttctcctCGGCCGTTCCACGGCCCGTATCGTC
      1        11   1267214   1267224   1267234   1267244

1266745   1266755   1266765   1266775   1266785   1266795
      GCCGCGCTAGGTTGGACGCTGTGCGGATCGTGGTGAGCAGTGCCACCAGAAATGCGGGTT
      ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
      GCCGCGCTAGGTTGGACGCTGTGCGGATCGTGGTGAGCAGTGCCACCAGAAATGCGGGTT
1267254   1267264   1267274   1267284   1267294   1267304

1266805   1266815   1266825   1266835   1266845   1266855
      CGTACACCTGTGTCAGCACCGGCAGCGCTGGATGCCGCGAGATTACACCGCCCCTCGCTG
      ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
      CGTACACCTGTGTCAGCACCGGCAGCGCTGGATGCCGCGAGATTACACCGCCCCTCGCTG
1267314   1267324   1267334   1267344   1267354   1267364

1266865   1266875   1266885   1266895   1266905   1266915
      GGCCCACGCCTGGGCCGGTGAACCCCGGCCCGCCCGCTGGCACCCTGCGAACCAGCCTGC
      ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
      GGCCCACGCCTGGGCCGGTGAACCCCGGCCCGCCCGCTGGCACCCTGCGAACCAGCCTGC
1267374   1267384   1267394   1267404   1267414   1267424
```

Where:

1-st line is the header:

```
[DD] Sequence:        1(      14), S:        726.8, L:  4411529 emb|AL123456|
MTBH37RV Mycobacterium tuberculosis complete genome
```

[DD]	Target sequence in direct chain (D), query sequence in direct chain (D). Variants: [DR] - target sequence in direct chain (D), query sequence in reverse chain (R). [RD] - target sequence in reverse chain (R), query sequence in direct chain (D). [RR] - target sequence in reverse chain (R), query sequence in reverse chain (R).
Sequence: 1(14)	Order number of sequence from a query set which is submitted to alignment. In brackets is an order number for alignment of this sequence (if it resulted in more than one alignment). Variants: 4 - the fifth alignment of the fourth sequence from a set
S	Score of this alignment.
L	Length of this query sequence
emb|AL123456|MTBH37RV Mycobacterium tuberculosis complete genome	Name of this query sequence

Additional information about alignment:

```
Summ of block lengths: 176235, Alignment bounds:
```

```
On first  sequence: start   1266719, end   1442971, length 176253
On second sequence: start   1267228, end   1443483, length 176256
```

length	The length covered by alignment, on target and query sequences appropriately.

List of alignment blocks:

```
Block of alignment: 9
    1 P:   1266719   1267228 L:   10640, G:  99.98, W: 106350, S:178.608
    2 P:   1277360   1277868 L:    6697, G:  99.90, W:  66760, S:141.524
```

Block of alignment: 8 - Number of blocks in this alignment. Each line below defines an appropriate block. Detailed description of a line from this list is shown further:

```
    1 P:   1266719   1267228 L:   10640, G:  99.98, W: 106350, S:178.608
```

1	Block number.
P: 1266719 1267228	Positions of similarity block' start on target and query sequences accordingly.
L: 10640	Length of this similarity block.
G: 99.98	Homology of this similarity block.
W: 106350	Weight of this similarity block (the arithmetic sum of symbols' similarity calculated from the given similarity matrix).
S:178.608	Score of this similarity block.

Alignment:

```
1266704   1266704   1266705   1266715   1266725   1266735
      ---------------(..)tgggaccgccattgcCGGGCCGTTCCACGGCCCGTATCGTC
      ...............(..)...............|||||||||||||||||||||||||
      ttgaccgatgacccc(..)tgcgcggcttctcctCGGGCCGTTCCACGGCCCGTATCGTC
      1        11   1267214   1267224   1267234   1267244
```

1 line - Numbering of the target sequence.
2 line - The target sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.
3 line - Separator line. Separator line symbols: "|" - perfect coincidence between symbols. Figures means the degree of symbols' similarity. Vary from 0 up to 9. 0 - no similarity, 9 - maximal similarity.
4 line - Numbering of the query sequence.
5 line - The query sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.

Parameters:

Input	
Target sequence	Place your query file with nucleotide sequences.
Query sequence(s)	Place file with one ore more nucleotide sequences.
Output	
Result	Name of the output file.
Format	Output format: **List of alignment blocks coordinates (default)** **List of alignment blocks coordinates and blocks sequences** **Output alignment** **General alignment information** **General alignment information, blocks list and alignment**

Sort blocks	Sort regions of homology for "List of alignment blocks coordinates" value of "Output format" option : **Don't sort (default)** **Incremental on Target** **Incremental on Query** **Decremental by score** **Decremental by weight** **Decremental by length**
Flank type	Flank type: **Length** - Output for given amount of symbols in flank of alignment block. **All** - unlimited flank
Position number	Print additional strings with position number for target and query strings.
Numeration Offset	Numeration Offset: **Target** - Given value will be added to taget sequence numeration on output **Query** - Given value will be added to query sequence numeration on output
Special symbols:	Special symbols: **Homology** - Output symbol as separator lines between target and query, each line separator position shows similarity between target and query positions **Gap** - Use given simbol to print output gaps **Tailing Gap** - Use given simbol to print output flanking gaps in profile output, default: '-' **Line Tearing** - String used for displaying of big gaps in alignment.
Output string	Output for given amount of symbols in each line.
Unalignment info	Produce output information for sequences where no similarity found.
Perfect only	Output perfect and near-perfect alignment.
Preprocessing	
Remove	Remove: **PolyA** - Remove polyA tail from taget sequence. It is may be useful if target sequence is mRNA or EST. **PolyT** - Remove polyT head from taget sequence. It is may be useful if target sequence is complemented mRNA or EST. **Trailing N** - Remove trailing N symbols from both ends of target sequence.
Cut Sequence	Cut Sequence: **Start** - Search in target sequence from given position **End** - Search in target sequence to given position. "0" - get to end
Apply to chain	Search in target sequence is applied to reverse chain.
Options	
Base	Base: **Large genomes/contigs** **Typical genomes/contigs** **Small genomes/contigs**
Score method	Scoring methods for whole alignment: **No scoring the alignment (default)** **By probability of the best block in alignment** **By probability of the summary of all blocks**

	Blast-like (in SD units) **Blast-like (in probability units)**
Threshold	If alignment has score less then given value then alignment is not printed.
Target chain(s)	Search in chain(s) in target: **In direct chain only** **In reverse chain only** **In both chains**
Fine adjustment	Fine adjustment of alignment blocks ends.
Multiply variants:	**Alternate variants** - Produce given best alternate variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best alternate variants of alignments. **Non-overlapped variants** - Produce given non-overlapped variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best non-overlapped alignments. **Different variants** - Produce given different variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce different variants of alignments.
Local alignment	Produce local alignment. Split alignment to several local alignments.
Split diagonal recursively	Split diagonal recursively (if possible).
Minimal required homology	Minimal required homology of the whole alignment.
Minimal required alignment length	Minimal required sum of alignment blocks length

MaliN

Multiple alignment for nucleotide sequences. Program is provided with viewer.

Parameters:

Input	
Sequences set	Place your set file nucleotide sequences in FASTA format
Output	
Result	Name of the output file
Options	
Scoring matrix	Select one of the standard pre-defined matrix.
Gap Initiation penalty	Gap Initiation penalty in average match units
Gap Continuation penalty	Gap Continuation penalty in average match units
Match score	Match score, if Single-score scoring chosen (Similarity scoring only)
Mismatch penalty	Mismatch penalty, if Single-score scoring chosen

MaliP

Multiple alignment for protein sequences. Program is provided with viewer.

Parameters:

Input	
Sequences set	Place your set file nucleotide sequences in FASTA format

Output	
Result	Name of the output file
Options	
Scoring matrix	Select one of the standard pre-defined matrix.
Gap Initiation penalty	Gap Initiation penalty in average match units
Gap Continuation penalty	Gap Continuation penalty in average match units
Match score	Match score, if Single-score scoring chosen (Similarity scoring only)
Mismatch penalty	Mismatch penalty, if Single-score scoring chosen

ProtMap

New Fast Tool for Aligning Proteins with Genome and Accurately Reconstructing Exon-intron Gene Structure

ProtMap program maps a set of protein sequences to a genomic sequence, producing gene structures and corresponding alignments of coding exons with the similar or identical protein queries. **ProtMap** uses a genomic sequence and a set of protein sequences as its input data, and reconstructs gene structure based on protein identity or homology, in contrast to a set of unordered alignment fragments generated by Blast. The program is very fast, and it produces gene structures similar to those of Genewise program, which is hundreds times slower (see Table 1 for speed comparison). Accuracy can be further significantly improved by use of **Fgenesh+** on ProtMap output: see Table 2 fro accuracy comparison).

ProtMap is used as a part of Softberry automatic genome annotation pipeline, **Fgenesh++C**. We also use it for generating putative gene models for genefinding parameters training on new genomes, for which few or no known genes are available. ProtMap is also very useful for finding pseudogenes as corrupted gene structures that map to known protein sequences.

Figure 1. Example of mapping a protein sequence to human chromosome 19.

```
L:3000000    Sequence Chr19 [cut:1 3000000]
[DD] Sequence:       1(      1), S:      105.56, L:1739
IPI:IPI00170643.1|SWISS-PROT:Q8TEK3-1 Tax_Id=9606 Splice isoform 2 of Q8TEK3
Summ of block lengths: 1284, Alignment bounds:
On first  sequence: start   2146727, end   2167197, length 20471
On second sequence: start       263, end      1682, length 1420
Blocks of alignment: 21
   1 E: 2146727      70 [ca GT] P: 2146727     263 L: 23, G: 101.574  S:14.75
   2 E: 2147573     107 [AG GT] P: 2147575     287 L: 35, G: 103.465, S:18.56
   3 E: 2148934      42 [AG GT] P: 2148934     322 L: 14, G: 103.043, S:11.68
   4 E: 2150399     111 [AG GT] P: 2150399     336 L: 37, G: 102.130, S:18.82
   5 E: 2150620     235 [AG GT] P: 2150620     373 L: 78, G: 101.500, S:27.15
   6 E: 2151098     114 [AG GT] P: 2151100     452 L: 37, G: 106.924, S:19.76
   7 E: 2151750      92 [AG GT] P: 2151752     490 L: 30, G: 101.424, S:16.82
   8 E: 2153538     102 [AG GT] P: 2153538     520 L: 34, G: 100.496, S:17.73
   9 E: 2153848     138 [AG GT] P: 2153848     554 L: 46, G:  99.003, S:20.30
  10 E: 2154470     126 [AG GT] P: 2154470     600 L: 42, G: 101.283, S:19.87

         1        11   2146713   2146723   2146739   2146769
         gatcacagaggctgg(..)agtgtctgtgtttca?[GGRIVSSKPFAPLNFRINSRNLSg
         ---------------(..)evdhqlkerfanmke  GGRIVSSKPFAPLNFRINSRNLS-
       248       248       249       259       267       277

   2146797   2146806   2147558   2147568   2147581   2147611
         ]gtaagaaactctcat(..)ctgtggctcctgcag[acIGTIMRVVELSPLKGSVSWTGK
          ---------------(..)--------------- -dIGTIMRVVELSPLKGSVSWTGK
       286       286       286       286       289       299
```

```
2147641   2147671   2147686   2148919   2148926   2148937
      PVSYYLHTIDRTI]gtgagtatctcgctg(..)ctttcttctttttag[LENYFSSLKNP
      PVSYYLHTIDRTI ---------------(..)--------------- LENYFSSLKNP
    309       319       322       322       322       323

2148967   2148982   2150384   2150391   2150402   2150432
      KLR]gtaagtttgtgtgtt(..)ctgctctccttccag[EEQEAARRRQQRESKSNAATP
      KLR ---------------(..)--------------- EEQEAARRRQQRESKSNAATP
    333       336       336       336       337       347

2150462   2150492   2150513   2150523   2150609   2150619
      TKGPEGKVAGPADAPM]gtaaggccccagcct(..)ccttgtgtcctccag[DSGAEEEK
      TKGPEGKVAGPADAPM ---------------(..)--------------- DSGAEEEK
    357       367       373       373       373       373
```

Table 1. Speed of processing sequences by Prot_Map, Fgenesh+ and GeneWise.

	Fgenesh+	**Prot_map**	**GeneWise**
88 sequences of genes < 20 kb	~1 min	~1 min	~90 min
8 sequences of genes > 400000 kb	~1 min	~1 min	~1200 min

Table 2. Comparison of accuracy of gene identification programs: ab initio Fgenesh and prediction with protein support: Fgenesh+ , GeneWise and Prot_Map on a set of human genes using mouse or drosophila homologous proteins. Sn ex, Sensitivity on exon level (exact exon predictions); Sno ex, sensitivity with exon overlap; Sp ex, specificity, exon level; Sn nuc, seisitivity, nucleotides; Sp nuc, specificity, nucleotides; CC, correlation coefficient; %CG, percent of genes predicted completely correctly (no missing and no extra exons, and all exon boundaries are predicted exactly correctly).

Mouse homologs: 60% < similarity level < 80% - 1425 sequences

	Sn ex	**Sno ex**	**Sp ex**	**Sn nuc**	**Sp nuc**	**CC**	**%CG**
Fgenesh	83.4	90.9	86.8	93.2	94.9	0.937	30
Genewise	88.1	96.5	90.5	97.8	99.2	0.984	43
Fgenesh+	93.9	97.9	94.9	98.4	99.3	0.988	65
Prot_map	87.0	96.5	86.6	97.0	98.5	0.976	40

Drosophila homologs: similarity level > 80% - 66 sequences.

	Sn ex	**Sno ex**	**Sp ex**	**Sn nuc**	**Sp nuc**	**CC**	**CG%**
Fgenesh	90.5	93.8	95.1	97.9	96.9	0.950	55
Genewise	79.3	83.9	86.8	97.3	99.5	0.985	23

Fgenesh+	95.1	97.8	97.0	98.9	99.5	0.9914	70
Prot_map	86.4	95.3	88.1	97.6	99.0	0.982	41

Parameters:

Input	
Target sequence	Place your query file with nucleotide sequences in FASTA format
Query sequence(s)	Place your second file with protein sequences in FASTA format
Output	
Result	Name of the output file.
Format	Output format: **List of alignment blocks coordinates** **List of alignment blocks coordinates and blocks sequences** **Output alignment (default)** **General alignment information** **General alignment information, blocks list and alignment**
Sort blocks	Sort regions of homology for "List of alignment blocks coordinates" value of "Output format" option : **Don't sort (default)** **Incremental on Target** **Incremental on Query** **Decremental by score** **Decremental by weight** **Decremental by length**
Flank type	Flank type: **Length** - Output for given amount of symbols in flank of alignment block. **All** - unlimited flank
Position number	Print additional strings with position number for target and query strings.
Numeration Offset	Numeration Offset: **Target** - Given value will be added to taget sequence numeration on output **Query** - Given value will be added to query sequence numeration on output
Special symbols:	Special symbols: **Homology** - Output symbol as separator lines between target and query, each line separator position shows similarity between target and query positions **Gap** - Use given simbol to print output gaps **Tailing Gap** - Use given simbol to print output flanking gaps in profile output, default: '-' **Line Tearing** - String used for displaying of big gaps in alignment.
Output string	Output for given amount of symbols in each line.
Unalignment info	Produce output information for sequences where no similarity found.
Perfect only	Output perfect and near-perfect alignment.
Preprocessing	
Remove	Remove: **PolyA** - Remove polyA tail from taget sequence. It is may be useful if target sequence is mRNA or EST.

	PolyT - Remove polyT head from taget sequence. It is may be useful if target sequence is complemented mRNA or EST. **Trailing N** - Remove trailing N symbols from both ends of target sequence.
Cut Sequence	Cut Sequence: **Start** - Search in target sequence from given position **End** - Search in target sequence to given position. "0" - get to end
Apply to chain	Search in target sequence is applied to reverse chain.
Options	
Alignment accurancy	Alignment accurancy: **Weak (fast)** **Normal (slow)**
Mapping accurancy	Mapping accurancy: **Weak (fast)** **Normal (slow)**
Score method	Scoring methods for whole alignment: **No scoring the alignment (default)** **By probability of the best block in alignment** **By probability of the summary of all blocks** **Blast-like (in SD units)** **Blast-like (in probability units)**
Threshold	If alignment has score less then given value then alignment is not printed.
Fine adjustment	Fine adjustment of alignment blocks ends.
Multiply variants:	**Alternate variants** - Produce given best alternate variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best alternate variants of alignments. **Non-overlapped variants** - Produce given non-overlapped variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best non-overlapped alignments. **Different variants** - Produce given different variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce different variants of alignments.
Local alignment	Produce local alignment. Split alignment to several local alignments
Split diagonal recursively	Split diagonal recursively (if possible).
Use consensus only for target sequence	If target sequence is per-aligned profile then during alignment process will be used target sequence consensus instead profile
Use consensus only for query sequence	If query sequence is per-aligned profile then during alignment process will be used query sequence consensus instead profile
Don't check mapping result for validity	Don't check mapping result for validity
Maximal allowed intron length	Maximal allowed intron length

SeqMatch-N

Program for aligning two multimegabyte-size genome sequences using a sequential search for most significant similarity regions

Program is provided with viewer.

Example of output:

```
L:426             Sequence    Duck  alpha-D  globin  mRNA,  complete  cds.  vs
C:\Documents                            and                          Settings\My
Documents\MolQuestWorkSpace\example_data\SeqMatch-N\seq1.fa
Total 1 sequences produce 1 significant alignment(s).

[DD]       1, S:       20.989, L:       429  Equus zebra alpha 1 globin gene,
complete cds.
****************************************************************************
[DD] Sequence:        1(       1), S:       20.989, L:       429  Equus  zebra
alpha 1 globin gene, complete cds.
Summ of block lengths: 356, Alignment bounds:
On target  sequence: start         1, end       408, length 408
On query sequence: start           1, end       411, length 411
Block of alignment: 8
    1 P:         1         1 L:       1, G: 100.00, W:     10, S:1
    2 P:         2         5 L:      21, G:  80.95, W:    130, S:5.65813
    3 P:        40        43 L:     159, G:  71.07, W:    670, S:13.332
    4 P:       205       208 L:       6, G: 100.00, W:     60, S:3.67423
    5 P:       216       219 L:      12, G:  91.67, W:    100, S:4.93771
    6 P:       235       238 L:      78, G:  80.77, W:    480, S:11.2317
    7 P:       326       329 L:      71, G:  66.20, W:    230, S:7.90613
    8 P:       401       404 L:       8, G: 100.00, W:     80, S:4.38178
          1         8         18        28        38        48
          A---TGCTGACCGCCGAGGACAAGAagctcatcacgcagttgTGGGAGAAGGTGGCTGGC
          |...|||||0|0||||00|||||||.................|||000|||||0|00|||
          AtggTGCTGTCTGCCGCCGACAAGAccaacgtcaaggccgccTGGAGTAAGGTTGGCGGC
          1         11        21        31        41        51

         58         68        78        88        98        108
          CACCAGGAGGAATTCGGAAGTGAAGCTCTGCAGAGGATGTTCCTCGCCTACCCCCAGACC
          0||000|00||0||0||0000||0||0||00|||||||||||||0|0||0||||000|||
          AACGCTGGCGAGTTTGGCGCAGAGGCCCTAGAGAGGATGTTCCTGGGCTTCCCCACCACC
         61         71        81        91        101       111

....
```

Where:

1-st line is the header:

```
[DD] Sequence:        1(       1), S:       20.989, L:       429  Equus  zebra
alpha 1 globin gene, complete cds.
```

[DD]	Target sequence in direct chain (D), query sequence in direct chain (D). Variants: [DR] - target sequence in direct chain (D), query sequence in reverse chain (R). [RD] - target sequence in reverse chain (R), query sequence in direct chain (D). [RR] - target sequence in reverse chain (R), query sequence in reverse chain (R).
Sequence: 1(1)	Order number of sequence from a query set which is submitted to alignment. In brackets is an order number for alignment of this sequence (if it resulted in more than one alignment). Variants: 4(5) - the fifth alignment of the fourth sequence from a set
S	Score of this alignment.

L	Length of this query sequence
Equus zebra alpha 1 globin gene, complete cds	Name of this query sequence

Additional information about alignment:

```
Summ of block lengths: 356, Alignment bounds:
On target  sequence: start         1, end       408, length 408
On query sequence: start           1, end       411, length 411
```

length	The length covered by alignment, in target and query sequences appropriately.

List of alignment blocks:

```
Block of alignment: 8
    1 P:         1         1 L:       1, G: 100.00, W:     10, S:1
    2 P:         2         5 L:      21, G:  80.95, W:    130, S:5.65813
```

Block of alignment: 8 - Number of blocks in this alignment. Each line below defines an appropriate block. Detailed description of a line from this list is shown further:

```
    1 P:         1         1 L:       1, G: 100.00, W:     10, S:1
```

1	Block number.
P: 1 1	Positions of similarity block' start in target and query sequences appropriately. In this case - from the first position in both sequences.
L: 1	Length of this similarity block.
G: 100.00	Homology of this similarity block.
W: 10	Weight of this similarity block (the arithmetic sum of symbols' similarity calculated from the given similarity matrix).
S:1	Score of this similarity block.

Alignment:

```
1         8        18        28        38        48
A---TGCTGACCGCCGAGGACAAGAagctcatcacgcagttgTGGGAGAAGGTGGCTGGC
|...|||||0|0||||00|||||||.................|||000|||||0|00|||
AtggTGCTGTCTGCCGCCGACAAGAccaacgtcaaggccgccTGGAGTAAGGTTGGCGGC
1        11        21        31        41        51
```

1 line - Numbering of the target sequence.
2 line - The target sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.
3 line - Separator line. Separator line symbols: "|" - perfect coincidence between symbols. Figures means the degree of symbols' similarity. Vary from 0 up to 9. 0 - no similarity, 9 - maximal similarity.
4 line - Numbering of the query sequence.
5 line - The query sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.
Parameters:

Input	
Target sequence	Place your query file with nucleotide sequences.
Query sequence(s)	Place file with one ore more nucleotide sequences.

Format	Input file format: **Packed** - Packed format **Fasta** - Fasta format
Output	
Result	Name of the output file.
Format	Output format: **List of alignment blocks coordinates** **List of alignment blocks coordinates and blocks sequences** **Output alignment (default)** **General alignment information** **General alignment information, blocks list and alignment**
Sort blocks	Sort regions of homology for "List of alignment blocks coordinates" value of "Output format" option : **Don't sort (default)** **Incremental on Target** **Incremental on Query** **Decremental by score** **Decremental by weight** **Decremental by length**
Flank type	Flank type: **Length** - Output for given amount of symbols in flank of alignment block. **All** - unlimited flank
Position number	Print additional strings with position number for target and query strings.
Numeration Offset	Numeration Offset: **Target** - Given value will be added to taget sequence numeration on output **Query** - Given value will be added to query sequence numeration on output
Special symbols:	Special symbols: **Homology** - Output symbol as separator lines between target and query, each line separator position shows similarity between target and query positions **Gap** - Use given simbol to print output gaps **Tailing Gap** - Use given simbol to print output flanking gaps in profile output, default: '-' **Line Tearing** - String used for displaying of big gaps in alignment.
Output string	Output for given amount of symbols in each line.
Unalignment info	Produce output information for sequences where no similarity found.
Perfect only	Output perfect and near-perfect alignment.
Graphic data	Name of the output binary t-file.
Preprocessing	
Remove	Remove: **PolyA** - Remove polyA tail from taget sequence. It is may be useful if target sequence is mRNA or EST. **PolyT** - Remove polyT head from taget sequence. It is may be useful if target sequence is complemented mRNA or EST. **Trailing N** - Remove trailing N symbols from both ends of target sequence.
Cut Sequence	Cut Sequence: **Start** - Search in target sequence from given position **End** - Search in target sequence to given position. "0" - get to end
Apply to chain	Search in target sequence is applied to reverse chain.
Options	

Precision	Precision: **Rough alignment (fast)** **Fast alignment (slow)**
Score method	Scoring methods for whole alignment: **No scoring the alignment (default)** **By probability of the best block in alignment** **By probability of the summary of all blocks** **Blast-like (in SD units)** **Blast-like (in probability units)**
Threshold	If alignment has score less then given value then alignment is not printed.
Target chain(s)t	Search in chain(s) in target: **In direct chain only** **In reverse chain only** **In both chains**
Fine adjustment	Fine adjustment of alignment blocks ends.
Multiply variants:	**Alternate variants** - Produce given best alternate variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best alternate variants of alignments. **Non-overlapped variants** - Produce given non-overlapped variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best non-overlapped alignments. **Different variants** - Produce given different variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce different variants of alignments.
Local alignment	Produce local alignment. Split alignment to several local alignments.
Split diagonal recursively	Split diagonal recursively (if possible).
Restrictions	**Target:** **By length** - Alignment region on target sequence does not exceed given length. **By multiplier** - Alignment region on target sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on target sequence does not exceed length of query sequence plus N. **Query:** **By length** - Alignment region on query sequence does not exceed given length. **By multiplier** - Alignment region on query sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on query sequence does not exceed length of query sequence plus N.

SeqMatchNW-N

The program implements Needleman-Wunsch algorithm to produce a global alignment of two nucleotide sequences. The approach is described in "A general method applicable to the search for similarities in the amino acid sequence of two proteins", J Mol Biol. 48(3):443-53. The Needleman-Wunsch algorithm uses dynamic programming, and is guaranteed to find the alignment with the maximum score with respect to the scoring system being used (which includes the substitution matrix and the gap-scoring scheme.

Program is provided with viewer.

Example of output:

```
L:999        Sequence gi|1418273|gb|U60902.1|OCU60902 Otolemur crassicaudatus
epsilon-, gamma-, delta-, and beta-globin genes, complete cds, and eta-globin
pseudogene
            vs            C:\Documents            and            Settings\My
Documents\MolQuestWorkSpace\example_data\SeqMatchNW-N\1\seq1.fa
Total 1 sequences produce 1 significant alignment(s).

[DD]       1, S:      14.962, L:      292 gi|455025|gb|U01317.1|HUMHBB Human
beta globin region on chromosome 11
****************************************************************************
[DD] Sequence:       1(      1), S:      14.962, L:      292 gi|455025|gb|
U01317.1|HUMHBB Human beta globin region on chromosome 11
Summ of block lengths: 251, Alignment bounds:
On first  sequence: start         1, end       940, length 940
On second sequence: start         2, end       292, length 291
Block of alignment: 37
    1 P:         1         2 L:       1, G: 100.00, W:      5, S:1
    2 P:        33         3 L:       4, G: 100.00, W:     20, S:2.82843
    3 P:        41         7 L:       4, G: 100.00, W:     20, S:2.82843
    4 P:        58        11 L:       3, G: 100.00, W:     15, S:2.32379
    5 P:       101        14 L:       7, G:  71.43, W:     17, S:2.50185
    6 P:       117        26 L:      13, G:  76.92, W:     38, S:4.02492
    7 P:       141        39 L:       3, G: 100.00, W:     15, S:2.32379
    8 P:       149        42 L:       3, G: 100.00, W:     15, S:2.32379
    9 P:       168        55 L:       9, G:  77.78, W:     27, S:3.30748
   10 P:       201        64 L:      13, G:  61.54, W:     20, S:2.83235
   11 P:       231        77 L:       4, G: 100.00, W:     20, S:2.82843
   12 P:       245        81 L:       3, G: 100.00, W:     15, S:2.32379
   13 P:       255        84 L:       4, G: 100.00, W:     20, S:2.82843
   14 P:       273        88 L:       8, G:  75.00, W:     22, S:2.92119
   15 P:       290        98 L:       8, G:  62.50, W:     13, S:2.19089
   16 P:       304       106 L:      11, G:  90.91, W:     46, S:4.64372
   17 P:       320       121 L:      10, G:  70.00, W:     23, S:3
   18 P:       346       139 L:       9, G:  77.78, W:     27, S:3.30748
   19 P:       368       148 L:       6, G:  83.33, W:     21, S:2.85774
   20 P:       378       154 L:      10, G:  80.00, W:     32, S:3.66667
   21 P:       392       164 L:       4, G: 100.00, W:     20, S:2.82843
   22 P:       411       171 L:       8, G:  75.00, W:     22, S:2.92119
   23 P:       426       179 L:       9, G:  66.67, W:     18, S:2.61116
   24 P:       467       188 L:      10, G:  90.00, W:     41, S:4.33333
   25 P:       482       198 L:       5, G:  80.00, W:     16, S:2.4004
   26 P:       502       203 L:       3, G: 100.00, W:     15, S:2.32379
   27 P:       515       207 L:      12, G:  83.33, W:     42, S:4.32049
   28 P:       547       226 L:      12, G:  75.00, W:     33, S:3.70328
   29 P:       621       238 L:       7, G:  85.71, W:     26, S:3.27165
   30 P:       641       245 L:       7, G:  71.43, W:     17, S:2.50185
   31 P:       653       252 L:       3, G: 100.00, W:     15, S:2.32379
   32 P:       706       255 L:       6, G:  83.33, W:     21, S:2.85774
   33 P:       727       261 L:      17, G:  70.59, W:     40, S:4.10605
   34 P:       888       278 L:       5, G:  80.00, W:     16, S:2.4004
   35 P:       907       283 L:       5, G: 100.00, W:     25, S:3.27327
   36 P:       929       288 L:       2, G: 100.00, W:     10, S:1.73205
   37 P:       938       290 L:       3, G: 100.00, W:     15, S:2.32379
         1 -AttaatagttgacagggatttacactaatgttATTCatcaTAATatgggatgtatcgCT
           .|...............................||||....||||.............||
         1 gA-------------------------------ATTC----TAAT-------------CT

        60 Cattgttgtttatttg(..)gaagaaaagttaaatCATTTCAttctttgtgAAAGACATC
           |...............(..)...............|0|0|||.........|0||0||0|
        13 C---------------(..)---------------CCTCTCAaccct----ACAGTCACC

       126 CATTaacccaccctcTGGatcacTATgctttagcagtttcaaTGTAGGCTAgtaagcctg
           ||||...........|||.....|||................|||0|0|||.........
        35 CATT-----------TGG-----TATattaaagatg------TGTTGTCTA---------

....
```

Where:

1-st line is the header:

```
[DD] Sequence:        1(       1), S:       14.962, L:       292 gi|455025|gb|
U01317.1|HUMHBB Human beta globin region on chromosome 11
```

[DD]	Target sequence in direct chain (D), query sequence in direct chain (D). Variants: [DR] - target sequence in direct chain (D), query sequence in reverse chain (R). [RD] - target sequence in reverse chain (R), query sequence in direct chain (D). [RR] - target sequence in reverse chain (R), query sequence in reverse chain (R).
Sequence: 1(1)	Order number of sequence from a query set which is submitted to alignment. In brackets is an order number for alignment of this sequence (if it resulted in more than one alignment). Variants: 4(5) - the fifth alignment of the fourth sequence from a set
S	Score of this alignment.
L	Length of this query sequence
gi|455025|gb|U01317.1|HUMHBB Human beta globin region on chromosome 11	Name of this query sequence

Additional information about alignment:

```
Summ of block lengths: 251, Alignment bounds:
On first  sequence: start         1, end       940, length 940
On second sequence: start         2, end       292, length 291
```

length	The length covered by alignment, in target and query sequences appropriately.

List of alignment blocks:

```
Block of alignment: 37
    1 P:         1         2 L:       1, G: 100.00, W:      5, S:1
    2 P:        33         3 L:       4, G: 100.00, W:     20, S:2.82843
```

Block of alignment: 37 - Number of blocks in this alignment. Each line below defines an appropriate block. Detailed description of a line from this list is shown further:

```
    2 P:        33         3 L:       4, G: 100.00, W:     20, S:2.82843
```

2	Block number.
P: 33 3	Positions of similarity block' start in target and query sequences appropriately.
L: 4	Length of this similarity block.
G: 100.00	Homology of this similarity block.

W: 20	Weight of this similarity block (the arithmetic sum of symbols' similarity calculated from the given similarity matrix).
S:2.82843	Score of this similarity block.

Alignment:

```
60 Cattgttgtttatttg(..)gaagaaaagttaaatCATTTCAttctttgtgAAAGACATC
   |...............(..)...............|0|0|||.........|0||0||0|
13 C---------------(..)---------------CCTCTCAaccct----ACAGTCACC
```

1 line - The target sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.

2 line - Separator line. Separator line symbols: "|" - perfect coincidence between symbols. Figures means the degree of symbols' similarity. Vary from 0 up to 9. 0 - no similarity, 9 - maximal similarity.

3 line - The query sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.

Parameters:

Input	
Target sequence	Place your query file with nucleotide sequences.
Query sequence(s)	Place file with one ore more nucleotide sequences.
Format	Input file format: **Packed** - Packed format **Fasta** - Fasta format
Output	
Result	Name of the output file.
Format	Output format: **List of alignment blocks coordinates** **List of alignment blocks coordinates and blocks sequences** **Output alignment (default)** **General alignment information** **General alignment information, blocks list and alignment**
Sort blocks	Sort regions of homology for "List of alignment blocks coordinates" value of "Output format" option : **Don't sort (default)** **Incremental on Target** **Incremental on Query** **Decremental by score** **Decremental by weight** **Decremental by length**
Flank type	Flank type: **Length** - Output for given amount of symbols in flank of alignment block. **All** - unlimited flank
Position number	Print additional strings with position number for target and query strings.
Numeration Offset	Numeration Offset: **Target** - Given value will be added to taget sequence numeration on output **Query** - Given value will be added to query sequence numeration on output
Special symbols:	Special symbols: **Homology** - Output symbol as separator lines between target and query, each line separator position shows similarity between target and query positions **Gap** - Use given simbol to print output gaps

	Tailing Gap - Use given simbol to print output flanking gaps in profile output, default: '-' **Line Tearing** - String used for displaying of big gaps in alignment.
Output string	Output for given amount of symbols in each line.
Unalignment info	Produce output information for sequences where no similarity found.
Perfect only	Output perfect and near-perfect alignment.
Graphic data	Name of the output binary t-file.
Preprocessing	
Remove	Remove: **PolyA** - Remove polyA tail from taget sequence. It is may be useful if target sequence is mRNA or EST. **PolyT** - Remove polyT head from taget sequence. It is may be useful if target sequence is complemented mRNA or EST. **Trailing N** - Remove trailing N symbols from both ends of target sequence.
Cut Sequence	Cut Sequence: **Start** - Search in target sequence from given position **End** - Search in target sequence to given position. "0" - get to end
Apply to chain	Search in target sequence is applied to reverse chain.
Options	
Scoring matrix	Select one of the standard pre-defined matrix.
Tail gap	Tail gap: **Alignment with tail gaps penalties** **Alignment without tail gaps penalties**
Gap Initiation penalty	Gap Initiation penalty in average match units.
Gap Continuation penalty	Gap Continuation penalty in average match units.
Match score	Match score, if Single-score scoring chosen (Similarity scoring only).
Mismatch penalty	Mismatch penalty, if Single-score scoring chosen.
Score method	Scoring methods for whole alignment: **No scoring the alignment (default)** **By probability of the best block in alignment** **By probability of the summary of all blocks** **Blast-like (in SD units)** **Blast-like (in probability units)**
Threshold	If alignment has score less then given value then alignment is not printed.
Target chain(s)	Search in chain(s) in target: **In direct chain only** **In reverse chain only** **In both chains**
Fine adjustment	Fine adjustment of alignment blocks ends.
Multiply variants:	**Alternate variants** - Produce given best alternate variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best alternate variants of alignments. **Non-overlapped variants** - Produce given non-overlapped variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best non-overlapped alignments. **Different variants** - Produce given different variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce different variants of alignments.

Local alignment	Produce local alignment. Split alignment to several local alignments.
Split diagonal recursively	Split diagonal recursively (if possible).
Restrictions	**Target:** **By length** - Alignment region on target sequence does not exceed given length. **By multiplier** - Alignment region on target sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on target sequence does not exceed length of query sequence plus N. **Query:** **By length** - Alignment region on query sequence does not exceed given length. **By multiplier** - Alignment region on query sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on query sequence does not exceed length of query sequence plus N.

SeqMatchNW-P

The program implements Needleman-Wunsch algorithm to produce a global alignment of two protein sequences. The approach is described in "A general method applicable to the search for similarities in the amino acid sequence of two proteins", J Mol Biol. 48(3):443-53. The Needleman-Wunsch algorithm uses dynamic programming, and is guaranteed to find the alignment with the maximum score with respect to the scoring system being used (which includes the substitution matrix and the gap-scoring scheme.

Program is provided with viewer.

Example of output:

```
L:153        Sequence MYOGLOBIN MAP TURTLE
vs.     19     Base     sequences     [C:\Documents     and     Settings\My
Documents\MolQuestWorkSpace\example_data\SeqMatchNW-P\seq1.set.fa].
Total 19 sequences produce 19 significant alignment(s).

[DD]       7, S:      28.714, L:      153 MYOGLOBIN CHICKEN
[DD]      17, S:       27.56, L:      153  MYOGLOBIN HUMAN
[DD]       9, S:      27.482, L:      153 MYOGLOBIN N.AMERICAN OPOSSUM
[DD]       5, S:      26.354, L:      153 MYOGLOBIN SADDLEBACK DOLPHIN
[DD]       8, S:      12.825, L:      146 HEMOGLOBIN BETA CHICKEN
[DD]      13, S:      12.696, L:      141 HEMOGLOBIN ALPHA NILE CROCODILE
[DD]      10, S:      12.388, L:      146 HEMOGLOBIN BETA N.AMERICAN OPOSSUM
[DD]       6, S:      12.271, L:      140 HEMOGLOBIN BETA EDIBLE FROG
[DD]      19, S:      12.226, L:      146  HEMOGLOBIN BETA HUMAN
[DD]      11, S:      11.998, L:      141 HEMOGLOBIN ALPHA BULLFROG
[DD]      14, S:      11.864, L:      141 HEMOGLOBIN ALPHA OSTRICH
[DD]      12, S:      11.533, L:      146 HEMOGLOBIN BETA NILE CROCODILE
[DD]       15, S:       11.521, L:       141   HEMOGLOBIN ALPHA EASTERN GRAY
KANGAROO
[DD]      18, S:      11.401, L:      141  HEMOGLOBIN ALPHA HUMAN
[DD]      16, S:      11.095, L:      142 HEMOGLOBIN ALPHA ABYSSINIAN HYRAX
[DD]       2, S:      9.9819, L:      161 HEMOGLOBIN I.PARASPONIA ANDERSONII
[DD]       1, S:      9.4062, L:      146 HEMOGLOBIN VITREOSCILLA SP.
[DD]       3, S:      8.1196, L:      153 LEGHEMOGLOBIN I. YELLOW LUPIN
[DD]       4, S:      6.8096, L:      143 LEGHEMOGLOBIN I.BROAD BEAN .
```

```
****************************************************************************
[DD] Sequence:         7(       1), S:       28.714, L:       153 MYOGLOBIN
CHICKEN
Summ of block lengths: 153, Alignment bounds:
On first  sequence: start         1, end       153, length 153
On second sequence: start         1, end       153, length 153
Block of alignment: 1
    1 P:         1         1 L:     153, G:  84.27, W: 874000, S:28.7142
        1 GLSDDEWHHVLGIWAKVEPDLSAHGQEVIIRLFQVHPETQERFAKFKNLKTIDELRSSEE
          ||||2||44||0||2|||1|552||4||55|||40||||05||0|||1|||05|662||5
        1 GLSDQEWQQVLTIWGKVEADIAGHGHEVLMRLFHDHPETLDRFDKFKGLKTPNEMKGSED

       61 VKKHGTTVLTALGRILKLKNNHEPELKPLAESHATKHKIPVKYLEFICEIIVKVIAEKHP
          4||||2||||1||6|||0|12||15|||||65|||||||||||||||1|7|7|||||||1
       61 LKKHGATVLTQLGKILKQKGQHESDLKPLAQTHATKHKIPVKYLEFISEVIIKVIAEKHA

      121 SDFGADSQAAMRKALELFRNDMASKYKEFGFQG
          5||||||||||6|||||||||||||||||||||
      121 ADFGADSQAAMKKALELFRNDMASKYKEFGFQG
[DD] Sequence:      17(      1), S:       27.56, L:      153  MYOGLOBIN HUMAN
Summ of block lengths: 153, Alignment bounds:
On first  sequence: start         1, end       153, length 153
On second sequence: start         1, end       153, length 153
Block of alignment: 1
    1 P:         1         1 L:     153, G:  81.13, W: 830000, S:27.5604
        1 GLSDDEWHHVLGIWAKVEPDLSAHGQEVIIRLFQVHPETQERFAKFKNLKTIDELRSSEE
          ||||0||40||17|2|||1|512|||||5||||50||||0|6|0|||4||50||665||5
        1 GLSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASED

       61 VKKHGTTVLTALGRILKLKNNHEPELKPLAESHATKHKIPVKYLEFICEIIVKVIAEKHP
          4||||2|||||||0|||0|14||1|5||||6||||||||||||||||1|0|75|512|||
       61 LKKHGATVLTALGGILKKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKHP

      121 SDFGADSQAAMRKALELFRNDMASKYKEFGFQG
          2|||||5|2||1|||||||2||||2|||4||||
      121 GDFGADAQGAMNKALELFRKDMASNYKELGFQG

....
```

Where:

1-st line is the header:

```
[DD] Sequence:         7(       1), S:       28.714, L:       153 MYOGLOBIN
CHICKEN
```

[DD]	No sence, used for output compatibility on nucleotide sequence alignment.
Sequence: 7(1)	Order number of sequence from a query set which is submitted to alignment. In brackets is an order number for alignment of this sequence (if it resulted in more than one alignment). Variants: 4(5) - the fifth alignment of the fourth sequence from a set.
S	Score of this alignment.
L	Length of this query sequence
MYOGLOBIN CHICKEN	Name of this query sequence

Additional information about alignment:

```
Summ of block lengths: 153, Alignment bounds:
On first  sequence: start         1, end       153, length 153
On second sequence: start         1, end       153, length 153
```

length	The length covered by alignment, in target and query sequences appropriately.

List of alignment blocks:

```
Block of alignment: 1
    1 P:         1         1 L:     153, G:  81.13, W: 830000, S:27.5604
```

Block of alignment: 1 - amount of blocks. Below each line corresponds to one block:

```
    1 P:         1         1 L:     153, G:  81.13, W: 830000, S:27.5604
```

1	Block number.
P: 1 1	Positions of similarity block' start in target and query sequences appropriately. In this case - from the first position in both sequences.
L: 153	Length of this similarity block.
G: 81.13	Homology of this similarity block.
W: 830000	Weight of this similarity block (the arithmetic sum of symbols' similarity calculated from the given similarity matrix).
S:27.5604	Score of this similarity block.

Alignment:

```
        1 GLSDDEWHHVLGIWAKVEPDLSAHGQEVIIRLFQVHPETQERFAKFKNLKTIDELRSSEE
          ||||2||44||0||2|||1|552||4||55|||40||||05||0|||1|||05|662||5
        1 GLSDQEWQQVLTIWGKVEADIAGHGHEVLMRLFHDHPETLDRFDKFKGLKTPNEMKGSED
```

1 line - The target sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.

2 line - Separator line. Separator line symbols: "|" - perfect coincidence between symbols. Figures means the degree of symbols' similarity. Vary from 0 up to 9. 0 - no similarity, 9 - maximal similarity.

3 line - The query sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.

Parameters:

Input	
Target sequence	Place your query file with protein sequences in FASTA format.
Query sequence(s)	Place input file with one ore more protein sequences in FASTA format.
Output	
Result	Name of the output file.
Format	Output format: **List of alignment blocks coordinates** **List of alignment blocks coordinates and blocks sequences** **Output alignment (default)** **General alignment information** **General alignment information, blocks list and alignment**
Sort blocks	Sort regions of homology for "List of alignment blocks coordinates" value of "Output format" option : **Don't sort (default)** **Incremental on Target** **Incremental on Query** **Decremental by score** **Decremental by weight** **Decremental by length**

Flank type	Flank type: **Length** - Output for given amount of symbols in flank of alignment block. **All** - unlimited flank
Position number	Print additional strings with position number for target and query strings.
Numeration Offset	Numeration Offset: **Target** - Given value will be added to taget sequence numeration on output **Query** - Given value will be added to query sequence numeration on output
Special symbols:	Special symbols: **Homology** - Output symbol as separator lines between target and query, each line separator position shows similarity between target and query positions **Gap** - Use given simbol to print output gaps **Tailing Gap** - Use given simbol to print output flanking gaps in profile output, default: '-' **Line Tearing** - String used for displaying of big gaps in alignment.
Output string	Output for given amount of symbols in each line.
Unalignment info	Produce output information for sequences where no similarity found.
Perfect only	Output perfect and near-perfect alignment.
Graphic data	Name of the output binary t-file.
Preprocessing	
Remove	Remove: **Trailing N** - Remove trailing N symbols from both ends of target sequence.
Cut Sequence	Cut Sequence: **Start** - Search in target sequence from given position **End** - Search in target sequence to given position. "0" - get to end
Apply to chain	Search in target sequence is applied to reverse chain.
Options	
Scoring matrix	Select one of the standard pre-defined matrix.
Tail gap	Tail gap: **Alignment with tail gaps penalties** **Alignment without tail gaps penalties**
Gap Initiation penalty	Gap Initiation penalty in average match units.
Gap Continuation penalty	Gap Continuation penalty in average match units.
Match score	Match score, if Single-score scoring chosen (Similarity scoring only).
Mismatch penalty	Mismatch penalty, if Single-score scoring chosen.
Score method	Scoring methods for whole alignment: **No scoring the alignment (default)** **By probability of the best block in alignment** **By probability of the summary of all blocks** **Blast-like (in SD units)** **Blast-like (in probability units)**
Threshold	If alignment has score less then given value then alignment is not printed.
Fine adjustment	Fine adjustment of alignment blocks ends.
Multiply variants:	**Alternate variants** - Produce given best alternate variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best alternate variants of alignments. **Non-overlapped variants** - Produce given non-overlapped variants of

	alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best non-overlapped alignments. **Different variants** - Produce given different variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce different variants of alignments.
Local alignment	Produce local alignment. Split alignment to several local alignments.
Split diagonal recursively	Split diagonal recursively (if possible).
Restrictions	**Target:** **By length** - Alignment region on target sequence does not exceed given length. **By multiplier** - Alignment region on target sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on target sequence does not exceed length of query sequence plus N. **Query:** **By length** - Alignment region on query sequence does not exceed given length. **By multiplier** - Alignment region on query sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on query sequence does not exceed length of query sequence plus N.
Score method	Scoring methods for whole alignment: **No scoring the alignment (default)** **By probability of the best block in alignment** **By probability of the summary of all blocks** **Blast-like (in SD units)** **Blast-like (in probability units)**
Threshold	If alignment has score less then given value then alignment is not printed.
Fine adjustment	Fine adjustment of alignment blocks ends.
Multiply variants:	**Alternate variants** - Produce given best alternate variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best alternate variants of alignments. **Non-overlapped variants** - Produce given non-overlapped variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best non-overlapped alignments. **Different variants** - Produce given different variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce different variants of alignments.
Local alignment	Produce local alignment. Split alignment to several local alignments.
Split diagonal recursively	Split diagonal recursively (if possible).
Restrictions	**Target:** **By length** - Alignment region on target sequence does not exceed given length. **By multiplier** - Alignment region on target sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on target sequence does not exceed length of query sequence plus N. **Query:** **By length** - Alignment region on query sequence does not exceed given

	length. **By multiplier** - Alignment region on query sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on query sequence does not exceed length of query sequence plus N.
Score method	Scoring methods for whole alignment: **No scoring the alignment (default)** **By probability of the best block in alignment** **By probability of the summary of all blocks** **Blast-like (in SD units)** **Blast-like (in probability units)**
Threshold	If alignment has score less then given value then alignment is not printed.
Fine adjustment	Fine adjustment of alignment blocks ends.

SeqMatch-P

Program for aligning two aminoacid sequences using a sequential search for most significant similarity regions.

Program is provided with viewer.

Example of output:

```
L:146         Sequence  HEMOGLOBIN BETA HUMAN
vs                C:\Documents                 and                 Settings\My
Documents\MolQuestWorkSpace\example_data\SeqMatch-P\seq1.fa
Total 1 sequences produce 1 significant alignment(s).

[DD]       1, S:      21.664, L:      146 HEMOGLOBIN BETA NILE CROCODILE
****************************************************************************
[DD] Sequence:       1(      1), S:      21.664, L:      146 HEMOGLOBIN BETA
NILE CROCODILE
Summ of block lengths: 124, Alignment bounds:
On first  sequence: start         7, end       146, length 140
On second sequence: start         7, end       146, length 140
Block of alignment: 6
    1 P:         7         7 L:       2, G: 100.51, W:     10, S:2.64676
    2 P:        14        14 L:       7, G:  83.27, W:     20, S:5.05147
    3 P:        24        24 L:      99, G:  78.57, W:    225, S:20.0317
    4 P:       128       128 L:       7, G:  94.76, W:     30, S:5.80101
    5 P:       137       137 L:       2, G:  92.46, W:      8, S:2.4219
    6 P:       140       140 L:       7, G:  82.12, W:     19, S:4.97651
        1 vhltpeEKsavtaLWGKVNVdevGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKV
          ......||.....||0||7|...|||||0|8|9||||07|9||7|||8|000|9|0|0||
        1 asfdphEKqligdLWHKVDVahcGGEALSRMLIVYPWKRRYFENFGDISNAQAIMHNEKV

       61 KAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGK
          7|||||||07|08070|||08800||0||7|||8|||||||||8|||79890|||0|90|
       61 QAHGKKVLASFGEAVCHLDGIRAHFANLSKLHCEKLHVDPENFKLLGDIIIIVLAAHYPK

      121 EFtppvqAAYQKVVagVAnALAHKYH
          8|.....|||||7|..||.|||07||
      121 DFglechAAYQKLVrqVAaALAAEYH
....
```

Where:

1-st line is the header:

```
[DD] Sequence:        1(      1), S:      21.664, L:      146 HEMOGLOBIN BETA
NILE CROCODILE
```

[DD]	No sence, used for output compatibility on nucleotide sequence alignment.
Sequence: 1(1)	Order number of sequence from a query set which is submitted to alignment. In brackets is an order number for alignment of this sequence (if it resulted in more than one alignment). Variants: 4(5) - the fifth alignment of the fourth sequence from a set
S	Score of this alignment.
L	Length of this query sequence
HEMOGLOBIN BETA NILE CROCODILE	Name of this query sequence

Additional information about alignment:

```
Summ of block lengths: 124, Alignment bounds:
On first  sequence: start         7, end       146, length 140
On second sequence: start         7, end       146, length 140
```

length	The length covered by alignment, in target and query sequences appropriately.

List of alignment blocks:

```
Block of alignment: 6
    1 P:        7         7 L:       2, G: 100.51, W:     10, S:2.64676
    2 P:       14        14 L:       7, G:  83.27, W:     20, S:5.05147
```

Block of alignment: 6 - Number of blocks in this alignment. Each line below defines an appropriate block. Detailed description of a line from this list is shown further:

```
    1 P:        7         7 L:       2, G: 100.51, W:     10, S:2.64676
```

1	Block number.
P: 7 7	Positions of similarity block' start in target and query sequences appropriately. In this case - from the seventh position in both sequences.
L: 2	Length of this similarity block.
G: 100.51	Homology of this similarity block.
W: 10	Weight of this similarity block (the arithmetic sum of symbols' similarity calculated from the given similarity matrix).
S:2.64676	Score of this similarity block.

Alignment:

```
        1 vhltpeEKsavtaLWGKVNVdevGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKV
          ......||.....||0||7|...|||||0|8|9||||07|9||7|||8|000|9|0|0||
        1 asfdphEKqligdLWHKVDVahcGGEALSRMLIVYPWKRRYFENFGDISNAQAIMHNEKV
```

1 line - The target sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.

2 line - Separator line. Separator line symbols: "|" - perfect coincidence between symbols. Figures means the degree of symbols' similarity. Vary from 0 up to 9. 0 - no similarity, 9 - maximal similarity.

3 line - The query sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.

Parameters:

Input	
Target sequence	Place your query file with protein sequences in FASTA format.
Query sequence(s)	Place input file with one ore more protein sequences in FASTA format.
Output	
Result	Name of the output file.
Format	Output format: **List of alignment blocks coordinates** **List of alignment blocks coordinates and blocks sequences** **Output alignment (default)** **General alignment information** **General alignment information, blocks list and alignment**
Sort blocks	Sort regions of homology for "List of alignment blocks coordinates" value of "Output format" option : **Don't sort (default)** **Incremental on Target** **Incremental on Query** **Decremental by score** **Decremental by weight** **Decremental by length**
Flank type	Flank type: **Length** - Output for given amount of symbols in flank of alignment block. **All** - unlimited flank
Position number	Print additional strings with position number for target and query strings.
Numeration Offset	Numeration Offset: **Target** - Given value will be added to taget sequence numeration on output **Query** - Given value will be added to query sequence numeration on output
Special symbols:	Special symbols: **Homology** - Output symbol as separator lines between target and query, each line separator position shows similarity between target and query positions **Gap** - Use given simbol to print output gaps **Tailing Gap** - Use given simbol to print output flanking gaps in profile output, default: '-' **Line Tearing** - String used for displaying of big gaps in alignment.
Output string	Output for given amount of symbols in each line.
Unalignment info	Produce output information for sequences where no similarity found.
Perfect only	Output perfect and near-perfect alignment.
Graphic data	Name of the output binary t-file.
Preprocessing	
Remove	Remove: **Trailing N** - Remove trailing N symbols from both ends of target sequence.
Cut Sequence	Cut Sequence: **Start** - Search in target sequence from given position **End** - Search in target sequence to given position. "0" - get to end
Apply to chain	Search in target sequence is applied to reverse chain.
Options	
Precision	Precision: **Rough (fast)** **Fine (slow)**

Score method	Scoring methods for whole alignment: **No scoring the alignment (default)** **By probability of the best block in alignment** **By probability of the summary of all blocks** **Blast-like (in SD units)** **Blast-like (in probability units)**
Threshold	If alignment has score less then given value then alignment is not printed.
Fine adjustment	Fine adjustment of alignment blocks ends.
Multiply variants:	**Alternate variants** - Produce given best alternate variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best alternate variants of alignments. **Non-overlapped variants** - Produce given non-overlapped variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best non-overlapped alignments. **Different variants** - Produce given different variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce different variants of alignments.
Local alignment	Produce local alignment. Split alignment to several local alignments.
Split diagonal recursively	Split diagonal recursively (if possible).
Restrictions	**Target:** **By length** - Alignment region on target sequence does not exceed given length. **By multiplier** - Alignment region on target sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on target sequence does not exceed length of query sequence plus N. **Query:** **By length** - Alignment region on query sequence does not exceed given length. **By multiplier** - Alignment region on query sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on query sequence does not exceed length of query sequence plus N.

SeqMatchSW-N

The program implements Smith-Waterman algorithm for performing local sequence alignment, finding similar regions between two nucleotide sequences. The approach is described in "Identification of Common Molecular Subsequences" , Journal of Molecular Biology, 147:195-197, 1981.The algorithm is a variation of the Needleman-Wunsch dynamic programming algorithm. It is guaranteed to find the optimal local alignment with respect to the scoring system being used (which includes the substitution matrix and the gap-scoring scheme).

Program is provided with viewer.

Example of output:

```
L:999         Sequence gi|1418273|gb|U60902.1|OCU60902 Otolemur crassicaudatus
epsilon-, gamma-, delta-, and beta-globin genes, complete cds, and eta-globin
pseudogene
```

```
           vs            C:\Documents            and            Settings\My
Documents\MolQuestWorkSpace\example_data\SeqMatchSW-N\1\seq1.fa
Total 1 sequences produce 1 significant alignment(s).

[DD]       1, S:      8.4023, L:      292 gi|455025|gb|U01317.1|HUMHBB Human
beta globin region on chromosome 11
**************************************************************************
[DD] Sequence:       1(      1), S:      8.4023, L:      292 gi|455025|gb|
U01317.1|HUMHBB Human beta globin region on chromosome 11
Summ of block lengths: 55, Alignment bounds:
On first  sequence: start       834, end       889, length 56
On second sequence: start       140, end       194, length 55
Block of alignment: 2
    1 P:       834       140 L:      12, G:  83.33, W:     42, S:4.32049
    2 P:       847       152 L:      43, G:  74.42, W:    116, S:7.31564
        1 attaatagttgacag(..)ttacattttctgagtTATACTTCCAGCtACTCAGGAGGCCG
          ...............(..)...............|0||0|||||||.|||000||||00|
      125 ---------------(..)gtggtggctcatgtcTGTAATTCCAGC-ACTGGAGAGGTAG

      860 AAATGGAGGATCCCTTGAGCTCAGGAGGTcaaggctgcagtgag(..)caaaaaactgc
          ||0|||||||000||||||||||0|||0|...............(..)...........
      165 AAGTGGAGGACTGCTTGAGCTCAAGAGTTtgatattatcctgga(..)gca--------

      996 tccg
          ....
      293 ----
....
```

Where:

1-st line is the header:

```
[DD] Sequence:       1(      1), S:      8.4023, L:      292 gi|455025|gb|
U01317.1|HUMHBB Human beta globin region on chromosome 11
```

[DD]	Target sequence in direct chain (D), query sequence in direct chain (D). Variants: [DR] - target sequence in direct chain (D), query sequence in reverse chain (R). [RD] - target sequence in reverse chain (R), query sequence in direct chain (D). [RR] - target sequence in reverse chain (R), query sequence in reverse chain (R).
Sequence: 1(1)	Order number of sequence from a query set which is submitted to alignment. In brackets is an order number for alignment of this sequence (if it resulted in more than one alignment). Variants: 4(5) - the fifth alignment of the fourth sequence from a set
S	Score of this alignment.
L	Length of this query sequence
gi|455025|gb|U01317.1|HUMHBB Human beta globin region on chromosome 11	Name of this query sequence

Additional information about alignment:

```
Summ of block lengths: 55, Alignment bounds:
On first  sequence: start       834, end       889, length 56
On second sequence: start       140, end       194, length 55
```

length	The length covered by alignment, in target and query sequences appropriately.

List of alignment blocks:

```
Block of alignment: 2
    1 P:       834       140 L:       12, G:  83.33, W:      42, S:4.32049
    2 P:       847       152 L:       43, G:  74.42, W:     116, S:7.31564
```

Block of alignment: 2 - amount of blocks. Below each line corresponds to one block:

```
    1 P:       834       140 L:       12, G:  83.33, W:      42, S:4.32049
```

1	Block number.
P: 834 140	Positions of similarity block' start in target and query sequences appropriately.
L: 12	Length of this similarity block.
G: 83.33	Homology of this similarity block.
W: 42	Weight of this similarity block (the arithmetic sum of symbols' similarity calculated from the given similarity matrix).
S:4.32049	Score of this similarity block.

Alignment:

```
      1 attaatagttgacag(..)ttacattttctgagtTATACTTCCAGCtACTCAGGAGGCCG
          ...............(..)...............|0||0|||||||.|||000||||00|
    125 ---------------(..)gtggtggctcatgtcTGTAATTCCAGC-ACTGGAGAGGTAG
```

1 line - Target sequence. Capital letters means blocks of similarity, lower case - not aligned regions.
2 line - Separator line. Separator line symbols: "|" - perfect coincidence between symbols.
Figures means the degree of symbols' similarity. Vary from 0 up to 9. 0 - no similarity, 9 - maximal similarity.
3 line - Query sequence. Capital letters means blocks of similarity, lower case - not aligned regions.
Parameters:

Input	
Target sequence	Place your query file with nucleotide sequences.
Query sequence(s)	Place file with one ore more nucleotide sequences.
Format	Input file format: **Packed** - Packed format **Fasta** - Fasta format
Output	
Result	Name of the output file.
Format	Output format: **List of alignment blocks coordinates** **List of alignment blocks coordinates and blocks sequences** **Output alignment (default)** **General alignment information** **General alignment information, blocks list and alignment**
Sort blocks	Sort regions of homology for "List of alignment blocks coordinates" value of "Output format" option : **Don't sort (default)** **Incremental on Target**

	Incremental on Query **Decremental by score** **Decremental by weight** **Decremental by length**
Flank type	Flank type: **Length** - Output for given amount of symbols in flank of alignment block. **All** - unlimited flank
Position number	Print additional strings with position number for target and query strings.
Numeration Offset	Numeration Offset: **Target** - Given value will be added to taget sequence numeration on output **Query** - Given value will be added to query sequence numeration on output
Special symbols:	Special symbols: **Homology** - Output symbol as separator lines between target and query, each line separator position shows similarity between target and query positions **Gap** - Use given simbol to print output gaps **Tailing Gap** - Use given simbol to print output flanking gaps in profile output, default: '-' **Line Tearing** - String used for displaying of big gaps in alignment.
Output string	Output for given amount of symbols in each line.
Unalignment info	Produce output information for sequences where no similarity found.
Perfect only	Output perfect and near-perfect alignment.
Graphic data	Name of the output binary t-file.
Preprocessing	
Remove	Remove: **PolyA** - Remove polyA tail from taget sequence. It is may be useful if target sequence is mRNA or EST. **PolyT** - Remove polyT head from taget sequence. It is may be useful if target sequence is complemented mRNA or EST. **Trailing N** - Remove trailing N symbols from both ends of target sequence.
Cut Sequence	Cut Sequence: **Start** - Search in target sequence from given position **End** - Search in target sequence to given position. "0" - get to end
Apply to chain	Search in target sequence is applied to reverse chain.
Options	
Scoring matrix	Select one of the standard pre-defined matrix.
Gap Initiation penalty	Gap Initiation penalty in average match units.
Gap Continuation penalty	Gap Continuation penalty in average match units.
Match score	Match score, if Single-score scoring chosen (Similarity scoring only).
Mismatch penalty	Mismatch penalty, if Single-score scoring chosen.
Score method	Scoring methods for whole alignment: **No scoring the alignment (default)** **By probability of the best block in alignment** **By probability of the summary of all blocks** **Blast-like (in SD units)** **Blast-like (in probability units)**
Threshold	If alignment has score less then given value then alignment is not printed.

Target chain(s)	Search in chain(s) in target: **In direct chain only** **In reverse chain only** **In both chains**
Fine adjustment	Fine adjustment of alignment blocks ends.
Multiply variants:	**Alternate variants** - Produce given best alternate variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best alternate variants of alignments. **Non-overlapped variants** - Produce given non-overlapped variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best non-overlapped alignments. **Different variants** - Produce given different variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce different variants of alignments.
Local alignment	Produce local alignment. Split alignment to several local alignments.
Split diagonal recursively	Split diagonal recursively (if possible).
Restrictions	**Target:** **By length** - Alignment region on target sequence does not exceed given length. **By multiplier** - Alignment region on target sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on target sequence does not exceed length of query sequence plus N. **Query:** **By length** - Alignment region on query sequence does not exceed given length. **By multiplier** - Alignment region on query sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on query sequence does not exceed length of query sequence plus N.

SeqMatchSW-P

The program implements Smith-Waterman algorithm for performing local sequence alignment, finding similar regions between two protein sequences. The approach is described in "Identification of Common Molecular Subsequences" , Journal of Molecular Biology, 147:195-197, 1981.The algorithm is a variation of the Needleman-Wunsch dynamic programming algorithm. It is guaranteed to find the optimal local alignment with respect to the scoring system being used (which includes the substitution matrix and the gap-scoring scheme).

Program is provided with viewer.

Example of output:

```
  L:153             Sequence  MYOGLOBIN  MAP  TURTLE  vs.  19  Base  sequences
[C:\Documents                          and                          Settings\My
Documents\MolQuestWorkSpace\example_data\SeqMatchSW-P\seq1.set.fa].
Total 19 sequences produce 19 significant alignment(s).

[DD]        7, S:      28.714, L:       153 MYOGLOBIN CHICKEN
[DD]       17, S:       27.56, L:       153  MYOGLOBIN HUMAN
[DD]        9, S:      27.482, L:       153 MYOGLOBIN N.AMERICAN OPOSSUM
[DD]        5, S:      26.354, L:       153 MYOGLOBIN SADDLEBACK DOLPHIN
```

```
[DD]       8, S:      12.825, L:      146 HEMOGLOBIN BETA CHICKEN
[DD]      13, S:      12.564, L:      141 HEMOGLOBIN ALPHA NILE CROCODILE
[DD]       6, S:      12.323, L:      140 HEMOGLOBIN BETA EDIBLE FROG
[DD]      10, S:      12.259, L:      146 HEMOGLOBIN BETA N.AMERICAN OPOSSUM
[DD]      19, S:      12.226, L:      146  HEMOGLOBIN BETA HUMAN
[DD]      11, S:      11.865, L:      141 HEMOGLOBIN ALPHA BULLFROG
[DD]      14, S:      11.713, L:      141 HEMOGLOBIN ALPHA OSTRICH
[DD]      15, S:      11.353, L:      141   HEMOGLOBIN ALPHA EASTERN GRAY
KANGAROO
[DD]      18, S:      11.235, L:      141  HEMOGLOBIN ALPHA HUMAN
[DD]      16, S:       10.87, L:      142 HEMOGLOBIN ALPHA ABYSSINIAN HYRAX
[DD]      12, S:      10.849, L:      146 HEMOGLOBIN BETA NILE CROCODILE
[DD]       2, S:      8.2676, L:      161 HEMOGLOBIN I.PARASPONIA ANDERSONII
[DD]       1, S:      7.6599, L:      146 HEMOGLOBIN VITREOSCILLA SP.
[DD]       3, S:      6.1534, L:      153 LEGHEMOGLOBIN I. YELLOW LUPIN
[DD]       4, S:      5.4138, L:      143 LEGHEMOGLOBIN I.BROAD BEAN .
****************************************************************************
[DD] Sequence:         7(       1), S:       28.714, L:       153 MYOGLOBIN
CHICKEN
Summ of block lengths: 153, Alignment bounds:
On first  sequence: start         1, end       153, length 153
On second sequence: start         1, end       153, length 153
Block of alignment: 1
    1 P:         1         1 L:     153, G:  84.27, W: 874000, S:28.7142
        1 GLSDDEWHHVLGIWAKVEPDLSAHGQEVIIRLFQVHPETQERFAKFKNLKTIDELRSSEE
          ||||2||44||0||2|||1|552||4||55|||40||||05||0|||1|||05|662||5
        1 GLSDQEWQQVLTIWGKVEADIAGHGHEVLMRLFHDHPETLDRFDKFKGLKTPNEMKGSED

       61 VKKHGTTVLTALGRILKLKNNHEPELKPLAESHATKHKIPVKYLEFICEIIVKVIAEKHP
          4||||2||||1||6|||0|12||15|||||65|||||||||||||||1|7|7|||||||1
       61 LKKHGATVLTQLGKILKQKGQHESDLKPLAQTHATKHKIPVKYLEFISEVIIKVIAEKHA

      121 SDFGADSQAAMRKALELFRNDMASKYKEFGFQG
          5||||||||||6|||||||||||||||||||||
      121 ADFGADSQAAMKKALELFRNDMASKYKEFGFQG
[DD] Sequence:      17(      1), S:       27.56, L:      153  MYOGLOBIN HUMAN
Summ of block lengths: 153, Alignment bounds:
On first  sequence: start         1, end       153, length 153
On second sequence: start         1, end       153, length 153
Block of alignment: 1
    1 P:         1         1 L:     153, G:  81.13, W: 830000, S:27.5604
        1 GLSDDEWHHVLGIWAKVEPDLSAHGQEVIIRLFQVHPETQERFAKFKNLKTIDELRSSEE
          ||||0||40||17|2|||1|512|||||5||||50||||0|6|0|||4||50||665||5
        1 GLSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASED

       61 VKKHGTTVLTALGRILKLKNNHEPELKPLAESHATKHKIPVKYLEFICEIIVKVIAEKHP
          4||||2|||||||0|||0|14||1|5||||6||||||||||||||||1|0|75|512|||
       61 LKKHGATVLTALGGILKKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKHP

      121 SDFGADSQAAMRKALELFRNDMASKYKEFGFQG
          2|||||5|2||1|||||||2||||2|||4||||
      121 GDFGADAQGAMNKALELFRKDMASNYKELGFQG
  ....
```

Where:

1-st line is the header:

```
[DD] Sequence:         7(       1), S:       28.714, L:       153 MYOGLOBIN
CHICKEN
```

[DD]	No sence, used for output compatibility on nucleotide sequence alignment.
Sequence: 7(7)	Order number of sequence from a query set which is submitted to

	alignment. In brackets is an order number for alignment of this sequence (if it resulted in more than one alignment). Variants: 4(5) - the fifth alignment of the fourth sequence from a set.
S	Score of this alignment.
L	Length of this query sequence
MYOGLOBIN CHICKEN	Name of this query sequence

Additional information about alignment:

```
Summ of block lengths: 153, Alignment bounds:
On first  sequence: start         1, end       153, length 153
On second sequence: start         1, end       153, length 153
```

length	The length covered by alignment, in target and query sequences appropriately.

List of alignment blocks:

```
Block of alignment: 1
    1 P:         1         1 L:     153, G:  84.27, W: 874000, S:28.7142
```

Block of alignment: 1 - amount of blocks. Below each line corresponds to one block:

```
    1 P:         1         1 L:     153, G:  84.27, W: 874000, S:28.7142
```

1	Block number.
P: 1 1	Positions of similarity block' start in target and query sequences appropriately. In this case - from the first position in both sequences.
L: 153	Length of this similarity block.
G: 84.27	Homology of this similarity block.
W: 874000	Weight of this similarity block (the arithmetic sum of symbols' similarity calculated from the given similarity matrix).
S:28.7142	Score of this similarity block.

Alignment:

```
        1 GLSDDEWHHVLGIWAKVEPDLSAHGQEVIIRLFQVHPETQERFAKFKNLKTIDELRSSEE
          ||||2||44||0||2|||1|552||4||55|||40||||05||0|||1|||05|662||5
        1 GLSDQEWQQVLTIWGKVEADIAGHGHEVLMRLFHDHPETLDRFDKFKGLKTPNEMKGSED
```

1 line - The target sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.

2 line - Separator line. Separator line symbols: "|" - perfect coincidence between symbols. Figures means the degree of symbols' similarity. Vary from 0 up to 9. 0 - no similarity, 9 - maximal similarity.

3 line - The query sequence itself. Capital letters correspond to blocks of similarity, lower case - not aligned regions.

Parameters:

Input	
Target sequence	Place your query file with protein sequences in FASTA format.
Query sequence(s)	Place input file with one ore more protein sequences in FASTA format.
Output	
Result	Name of the output file.
Format	Output format: **Don't sort (default)**

	Incremental on Target **Incremental on Query** **Decremental by score** **Decremental by weight** **Decremental by length**
Sort blocks	Sort regions of homology for "List of alignment blocks coordinates" value of "Output format" option : **Don't sort (default)** **Incremental sort by coordinates on target** **Incremental sort by coordinates on Query** **Decremental sort by alignment block score** **Decremental sort by alignment block weight** **Decremental sort by alignment block length**
Flank type	Flank type: **Length** - Output for given amount of symbols in flank of alignment block. **All** - unlimited flank
Position number	Print additional strings with position number for target and query strings.
Numeration Offset	Numeration Offset: **Target** - Given value will be added to taget sequence numeration on output **Query** - Given value will be added to query sequence numeration on output
Special symbols:	Special symbols: **Homology** - Output symbol as separator lines between target and query, each line separator position shows similarity between target and query positions **Gap** - Use given simbol to print output gaps **Tailing Gap** - Use given simbol to print output flanking gaps in profile output, default: '-' **Line Tearing** - String used for displaying of big gaps in alignment.
Output string	Output for given amount of symbols in each line.
Unalignment info	Produce output information for sequences where no similarity found.
Perfect only	Output perfect and near-perfect alignment.
Graphic data	Name of the output binary t-file.
Preprocessing	
Remove	Remove: **Trailing N** - Remove trailing N symbols from both ends of target sequence.
Cut Sequence	Cut Sequence: **Start** - Search in target sequence from given position **End** - Search in target sequence to given position. "0" - get to end
Apply to chain	Search in target sequence is applied to reverse chain.
Options	
Scoring matrix	Select one of the standard pre-defined matrix.
Gap Initiation penalty	Gap Initiation penalty in average match units.
Gap Continuation penalty	Gap Continuation penalty in average match units.
Match score	Match score, if Single-score scoring chosen (Similarity scoring only).
Mismatch penalty	Mismatch penalty, if Single-score scoring chosen.
Score method	Scoring methods for whole alignment: **No scoring the alignment (default)** **By probability of the best block in alignment**

	By probability of the summary of all blocks **Blast-like (in SD units)** **Blast-like (in probability units)**
Threshold	If alignment has score less then given value then alignment is not printed.
Fine adjustment	Fine adjustment of alignment blocks ends.
Multiply variants:	**Alternate variants** - Produce given best alternate variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best alternate variants of alignments. **Non-overlapped variants** - Produce given non-overlapped variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce given best non-overlapped alignments. **Different variants** - Produce given different variants of alignments. "All" - Produce all alternate variants of alignments. "Number" - Produce different variants of alignments.
Local alignment	Produce local alignment. Split alignment to several local alignments.
Split diagonal recursively	Split diagonal recursively (if possible).
Restrictions	**Target:** **By length** - Alignment region on target sequence does not exceed given length. **By multiplier** - Alignment region on target sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on target sequence does not exceed length of query sequence plus N. **Query:** **By length** - Alignment region on query sequence does not exceed given length. **By multiplier** - Alignment region on query sequence does not exceed length of query sequence multiplied to N (N - is floting poin number). **By range** - Alignment region on query sequence does not exceed length of query sequence plus N.

Description of pre-defined matrix

ALTS910101 The PAM-120 matrix (Altschul, 1991)
LIT:1713145 PMID:2051488
Altschul, S.F.
Amino acid substitution matrices from an information theoretic perspective
J. Mol. Biol. 219, 555-565 (1991)

BENS940101 Log-odds scoring matrix collected in 6.4-8.7 PAM (Benner et al., 1994)
LIT:2023094 PMID:7700864
Benner, S.A., Cohen, M.A. and Gonnet, G.H.
Amino acid substitution during functionally constrained divergent evolution of protein sequences
Protein Engineering 7, 1323-1332 (1994) * extrapolated to 250 PAM

BENS940102 Log-odds scoring matrix collected in 22-29 PAM (Benner et al., 1994)
LIT:2023094 PMID:7700864
Benner, S.A., Cohen, M.A. and Gonnet, G.H.
Amino acid substitution during functionally constrained divergent

evolution of protein sequences
Protein Engineering 7, 1323-1332 (1994) * extrapolated to 250 PAM

BENS940103 Log-odds scoring matrix collected in 74-100 PAM (Benner et al., 1994)
LIT:2023094 PMID:7700864
Benner, S.A., Cohen, M.A. and Gonnet, G.H.
Amino acid substitution during functionally constrained divergent evolution of protein sequences
Protein Engineering 7, 1323-1332 (1994) * extrapolated to 250 PAM

BENS940104 Genetic code matrix (Benner et al., 1994)
LIT:2023094 PMID:7700864
Benner, S.A., Cohen, M.A. and Gonnet, G.H.
Amino acid substitution during functionally constrained divergent evolution of protein sequences
Protein Engineering 7, 1323-1332 (1994) * extrapolated to 250 PAM

CSEM940101 Residue replace ability matrix (Cserzo et al., 1994)
LIT:2022066 PMID:7966267
Cserzo, M., Bernassau, J.-M., Simon, I. and Maigret, B.
New alignment strategy for transmembrane proteins
J. Mol. Biol. 243, 388-396 (1994) * Diagonal elements are missing. * We use 1 as diagonal elements.

DAYM780301 Log odds matrix for 250 PAMs (Dayhoff et al., 1978) R
Dayhoff, M.O., Schwartz, R.M. and Orcutt, B.C.
A model of evolutionary change in proteins
In "Atlas of Protein Sequence and Structure", Vol.5, Suppl.3 (Dayhoff, M.O., ed.), National Biomedical Research Foundation, Washington, D.C., p.352 (1978)

FEND850101 Structure-Genetic matrix (Feng et al., 1985)
LIT:1107900 PMID:6100188
Feng, D.F., Johnson, M.S. and Doolittle, R.F.
Aligning amino acid sequences: comparison of commonly used methods
J. Mol. Evol. 21, 112-125 (1985)

FITW660101 Mutation values for the interconversion of amino acid pairs (Fitch, 1966)
PMID:5917736
Fitch, W.M.
An improved method of testing for evolutionary homology
J. Mol. Biol. 16, 9-16 (1966)

GEOD900101 Hydrophobicity scoring matrix (George et al., 1990)
PMID:2314281
George, D.G., Barker, W.C. and Hunt, L.T.
Mutation data matrix and its uses
Methods Enzymol. 183, 333-351 (1990)

GONG920101 The mutation matrix for initially aligning (Gonnet et al., 1992)
LIT:1813110 PMID:1604319

Gonnet, G.H., Cohen, M.A. and Benner, S.A.
Exhaustive matching of the entire protein sequence database
Science 256, 1443-1445 (1992)

GRAR740104 Chemical distance (Grantham, 1974)
LIT:2004143 PMID:4843792
Grantham, R.
Amino acid difference formula to help explain protein evolution
Science 185, 862-864 (1974)

HENS920101 BLOSUM45 substitution matrix (Henikoff-Henikoff, 1992)
LIT:1902106 PMID:1438297
Henikoff, S. and Henikoff, J.G.
Amino acid substitution matrices from protein blocks
Proc. Natl. Acad. Sci. USA 89, 10915-10919 (1992) * matrix in 1/3 Bit Units

HENS920102 BLOSUM62 substitution matrix (Henikoff-Henikoff, 1992)
LIT:1902106 PMID:1438297
Henikoff, S. and Henikoff, J.G.
Amino acid substitution matrices from protein blocks
Proc. Natl. Acad. Sci. USA 89, 10915-10919 (1992) * matrix in 1/3 Bit Units

HENS920103 BLOSUM80 substitution matrix (Henikoff-Henikoff, 1992)
LIT:1902106 PMID:1438297
Henikoff, S. and Henikoff, J.G.
Amino acid substitution matrices from protein blocks
Proc. Natl. Acad. Sci. USA 89, 10915-10919 (1992) * matrix in 1/3 Bit Units

JOHM930101 Structure-based amino acid scoring table (Johnson-Overington, 1993)
LIT:1923112 PMID:8411177
Johnson, M.S. and Overington, J.P.
A structural basis for sequence comparisons An evaluation of scoring methodologies
J. Mol. Biol. 233, 716-738 (1993)

JOND920103 The 250 PAM PET91 matrix (Jones et al., 1992)
LIT:1814076 PMID:1633570
Jones, D.T., Taylor, W.R. and Thornton, J.M.
The rapid generation of mutation data matrices from protein sequences
CABIOS 8, 275-282 (1992)

JOND940101 The 250 PAM transmembrane protein exchange matrix (Jones et al., 1994)
LIT:2006072 PMID:8112466
Jones, D.T., Taylor, W.R. and Thornton, J.M.
A mutation data matrix for transmembrane proteins
FEBS Lett. 339, 269-275 (1994)

KOLA920101 Conformational similarity weight matrix (Kolaskar-Kulkarni-Kale, 1992)
LIT:1806109 PMID:1538389
Kolaskar, A.S. and Kulkarni-Kale, U.
Sequence alignment approach to pick up conformationally similar protein fragments
J. Mol. Biol. 223, 1053-1061 (1992)

LEVJ860101 The secondary structure similarity matrix (Levin et al., 1986)
LIT:1210126 PMID:3743779
Levin, J.M., Robson, B. and Garnier, J.
An algorithm for secondary structure determination in proteins based on sequence similarity
FEBS Lett. 205, 303-308 (1986)

LUTR910101 Structure-based comparison table for outside other class (Luthy et al., 1991)
LIT:1712085 PMID:1881879
Luthy, R., McLachlan, A.D. and Eisenberg, D.
Secondary structure-based profiles: Use of structure-conserving scoring tables in searching protein sequence databases for structural similarities
Proteins 10, 229-239 (1991)

LUTR910102 Structure-based comparison table for inside other class (Luthy et al., 1991)
LIT:1712085 PMID:1881879
Luthy, R., McLachlan, A.D. and Eisenberg, D.
Secondary structure-based profiles: Use of structure-conserving scoring tables in searching protein sequence databases for structural similarities
Proteins 10, 229-239 (1991)

LUTR910103 Structure-based comparison table for outside alpha class (Luthy et al., 1991)
LIT:1712085 PMID:1881879
Luthy, R., McLachlan, A.D. and Eisenberg, D.
Secondary structure-based profiles: Use of structure-conserving scoring tables in searching protein sequence databases for structural similarities
Proteins 10, 229-239 (1991)

LUTR910104 Structure-based comparison table for inside alpha class (Luthy et al., 1991)
LIT:1712085 PMID:1881879
Luthy, R., McLachlan, A.D. and Eisenberg, D.
Secondary structure-based profiles: Use of structure-conserving scoring tables in searching protein sequence databases for structural similarities
Proteins 10, 229-239 (1991)

LUTR910105 Structure-based comparison table for outside beta class (Luthy et al., 1991)
LIT:1712085 PMID:1881879
Luthy, R., McLachlan, A.D. and Eisenberg, D.
Secondary structure-based profiles: Use of structure-conserving scoring

tables in searching protein sequence databases for structural similarities
Proteins 10, 229-239 (1991)

LUTR910106 Structure-based comparison table for inside beta class (Luthy et al., 1991)
LIT:1712085 PMID:1881879
Luthy, R., McLachlan, A.D. and Eisenberg, D.
Secondary structure-based profiles: Use of structure-conserving scoring tables in searching protein sequence databases for structural similarities
Proteins 10, 229-239 (1991)

LUTR910107 Structure-based comparison table for other class (Luthy et al., 1991)
LIT:1712085 PMID:1881879
Luthy, R., McLachlan, A.D. and Eisenberg, D.
Secondary structure-based profiles: Use of structure-conserving scoring tables in searching protein sequence databases for structural similarities
Proteins 10, 229-239 (1991)

LUTR910108 Structure-based comparison table for alpha helix class (Luthy et al., 1991)
LIT:1712085 PMID:1881879
Luthy, R., McLachlan, A.D. and Eisenberg, D.
Secondary structure-based profiles: Use of structure-conserving scoring tables in searching protein sequence databases for structural similarities
Proteins 10, 229-239 (1991)

LUTR910109 Structure-based comparison table for beta strand class (Luthy et al., 1991)
LIT:1712085 PMID:1881879
Luthy, R., McLachlan, A.D. and Eisenberg, D.
Secondary structure-based profiles: Use of structure-conserving scoring tables in searching protein sequence databases for structural similarities
Proteins 10, 229-239 (1991)

MCLA710101 The similarity of pairs of amino acids (McLachlan, 1971)
PMID:5167087
McLachlan, A.D.
Tests for comparing related amino-acid sequences cytochrome c and cytochrome c551
J. Mol. Biol. 61, 409-424 (1971) * (RR 9.)

MCLA720101 Chemical similarity scores (McLachlan, 1972)
PMID:5023183
McLachlan, A.D.
Repeating sequences and gene duplication in proteins
J. Mol. Biol. 64, 417-437 (1972)

MIYS930101 Base-substitution-protein-stability matrix (Miyazawa-Jernigan, 1993)
LIT:1913158 PMID:8506261
Miyazawa, S. and Jernigan, R.L.
A new substitution matrix for protein sequence searches based on

contact frequencies in protein structures
Protein Engineering 6, 267-278 (1993)

MIYT790101 Amino acid pair distance (Miyata et al., 1979)
LIT:0601606 PMID:439147
Miyata, T., Miyazawa, S. and Yasunaga, T.
Two types of amino acid substitutions in protein evolution
J. Mol. Evol. 12, 219-236 (1979)

MOHR870101 EMPAR matrix (Mohana Rao, 1987)
LIT:1304091 PMID:3570667
Mohana Rao, J.K.
New scoring matrix for amino acid residue exchanges based on residue characteristic physical parameters
Int. J. Peptide Protein Res. 29, 276-281 (1987)

NIEK910101 Structure-derived correlation matrix 1 (Niefind-Schomburg, 1991)
LIT:1713140 PMID:2051484
Niefind, K. and Schomburg, D.
Amino acid similarity coefficients for protein modeling and sequence alignment derived from main-chain folding angles
J. Mol. Biol. 219, 481-497 (1991)

NIEK910102 Structure-derived correlation matrix 2 (Niefind-Schomburg, 1991)
LIT:1713140 PMID:2051484
Niefind, K. and Schomburg, D.
Amino acid similarity coefficients for protein modeling and sequence alignment derived from main-chain folding angles
J. Mol. Biol. 219, 481-497 (1991)

OVEJ920101 STR matrix from structure-based alignments (Overington et al., 1992)
LIT:1811128 PMID:1304904
Overington, J., Donnelly, D., Johnson, M.S., Sali, A. and Blundell, T.L.
Environment-specific amino acid substitution tables: tertiary templates and prediction of protein folds
Protein Science 1, 216-226 (1992)

QU_C930101 Cross-correlation coefficients of preference factors main chain (Qu et al., 1993)
LIT:1906100 PMID:8381879
Qu, C., Lai, L., Xu, X. and Tang, Y.
Phyletic relationships of protein structures based on spatial prefernce of residues
J. Mol. Evol. 36, 67-78 (1993)

QU_C930102 Cross-correlation coefficients of preference factors side chain (Qu et al., 1993)
LIT:1906100 PMID:8381879
Qu, C., Lai, L., Xu, X. and Tang, Y.
Phyletic relationships of protein structures based on spatial prefernce of residues

J. Mol. Evol. 36, 67-78 (1993)

QU_C930103 The mutant distance based on spatial preference factor (Qu et al., 1993)
LIT:1906100 PMID:8381879
Qu, C., Lai, L., Xu, X. and Tang, Y.
Phyletic relationships of protein structures based on spatial prefernce of residues
J. Mol. Evol. 36, 67-78 (1993)

RISJ880101 Scoring matrix (Risler et al., 1988)
LIT:1505154 PMID:3221397
Risler, J.L., Delorme, M.O., Delacroix, H. and Henaut, A.
Amino acid substitutions in structurally related proteins A pattern recognition approach Determination of a new and efficient scoring matrix
J. Mol. Biol. 204, 1019-1029 (1988)

TUDE900101 isomorphicity of replacements (Tudos et al., 1990)
LIT:1616619 PMID:2279846
Tudos, E., Cserzo, M. and Simon, I.
Predicting isomorphic residue replacements for protein design
Int. J. Peptide Protein Res. 36, 236-239 (1990) * Diagonal elements are missing. * We use 100 as diagonal elements.

AZAE970101 The single residue substitution matrix from interchanges of spatially neighbouring residues (Azarya-Sprinzak et al., 1997)
PMID:9488136
Azarya-Sprinzak, E., Naor, D., Wolfson, H.J. and Nussinov, R.
Interchanges of spatially neighbouring residues in structurally conserved environments.
Protein Engineering 10, 1109-1122 (1997)

AZAE970102 The substitution matrix derived from spatially conserved motifs (Azarya-Sprinzak et al., 1997)
PMID:9488136
Azarya-Sprinzak, E., Naor, D., Wolfson, H.J. and Nussinov, R.
Interchanges of spatially neighbouring residues in structurally conserved environments.
Protein Engineering 10, 1109-1122 (1997)

RIER950101 Hydrophobicity scoring matrix (Riek et al., 1995)
PMID:7715195
Riek, R.P., Handschumacher, M.D., Sung, S.S., Tan, M., Glynias, M.J., Schluchter, M.D., Novotny, J. and Graham, R.M.
Evolutionary conservation of both the hydrophilic and hydrophobic nature of transmembrane residues.
J. Theor. Biol. 172, 245-258 (1995)

WEIL970101 WAC matrix constructed from amino acid comparative profiles (Wei et al., 1997)
PMID:9390315

Wei, L., Altman, R.B. and Chang, J.T.
Using the radial distributions of physical features to compare amino acid environments and align amino acid sequences.
Pac. Symp. Biocomput. 1997 5, 465-476 (1997)

WEIL970102 Difference matrix obtained by subtracting the BLOSUM62 from the WAC matrix (Wei et al., 1997)
PMID:9390315
Wei, L., Altman, R.B. and Chang, J.T.
Using the radial distributions of physical features to compare amino acid environments and align amino acid sequences.
Pac. Symp. Biocomput. 1997 5, 465-476 (1997)

MEHP950101 (Mehta et al., 1995)
LIT:2213135 PMID:8580842
Mehta, P.K., Heringa, J. and Argos, P.
A simple and fast approach to prediction of protein secondary structure from multiply aligned sequences with accuracy above 70%
Protein Science 4, 2517-2525 (1995)

MEHP950102 (Mehta et al., 1995)
LIT:2213135 PMID:8580842
Mehta, P.K., Heringa, J. and Argos, P.
A simple and fast approach to prediction of protein secondary structure from multiply aligned sequences with accuracy above 70%
Protein Science 4, 2517-2525 (1995)

MEHP950103 (Mehta et al., 1995)
LIT:2213135 PMID:8580842
Mehta, P.K., Heringa, J. and Argos, P.
A simple and fast approach to prediction of protein secondary structure from multiply aligned sequences with accuracy above 70%
Protein Science 4, 2517-2525 (1995)

KAPO950101 (Kapp et al., 1995)
LIT:2124159 PMID:8535255
Kapp, O.H., Moens, L., Vanfleteren, J., Trotman, C.N., Suzuki, T. and Vinogradov, S.N.
Alignment of 700 globin sequences: extent of amino acid substitution and its correlation with variation in volume
Protein Science 4, 2179-2190 (1995)

VOGG950101 (Vogt et al., 1995)
LIT:2114150 PMID:7602593
Vogt G, Etzold T, Argos P
An assessment of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited
J. Mol. Biol. 249, 816-831 (1995)

KOSJ950101 Context-dependent optimal substitution matrices for exposed helix (Koshi-Goldstein, 1995)

LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950102 Context-dependent optimal substitution matrices for exposed beta (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950103 Context-dependent optimal substitution matrices for exposed turn (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950104 Context-dependent optimal substitution matrices for exposed coil (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950105 Context-dependent optimal substitution matrices for buried helix (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950106 Context-dependent optimal substitution matrices for buried beta (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950107 Context-dependent optimal substitution matrices for buried turn (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950108 Context-dependent optimal substitution matrices for buried coil (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.

Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950109 Context-dependent optimal substitution matrices for alpha helix (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950110 Context-dependent optimal substitution matrices for beta sheet (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950111 Context-dependent optimal substitution matrices for turn (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950112 Context-dependent optimal substitution matrices for coil (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950113 Context-dependent optimal substitution matrices for exposed residues (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950114 Context-dependent optimal substitution matrices for buried residues (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

KOSJ950115 Context-dependent optimal substitution matrices for all residues (Koshi-Goldstein, 1995)
LIT:2124140 PMID:8577693
Koshi, J.M. and Goldstein, R.A.
Context-dependent optimal substitution matrices.
Protein Engineering 8, 641-645 (1995)

OVEJ920102 Environment-specific amino acid substitution matrix for alpha residues (Overington et al., 1992)
LIT:1811128 PMID:1304904
Overington, J., Donnelly, D., Johnson, M.S., Sali, A. and Blundell, T.L.
Environment-specific amino acid substitution tables: tertiary templates and prediction of protein folds
Protein Science 1, 216-226 (1992)

OVEJ920103 Environment-specific amino acid substitution matrix for beta residues (Overington et al., 1992)
LIT:1811128 PMID:1304904
Overington, J., Donnelly, D., Johnson, M.S., Sali, A. and Blundell, T.L.
Environment-specific amino acid substitution tables: tertiary templates and prediction of protein folds
Protein Science 1, 216-226 (1992)

OVEJ920104 Environment-specific amino acid substitution matrix for accessible residues (Overington et al., 1992)
LIT:1811128 PMID:1304904
Overington, J., Donnelly, D., Johnson, M.S., Sali, A. and Blundell, T.L.
Environment-specific amino acid substitution tables: tertiary templates and prediction of protein folds
Protein Science 1, 216-226 (1992)

OVEJ920105 Environment-specific amino acid substitution matrix for inaccessible residues (Overington et al., 1992)
LIT:1811128 PMID:1304904
Overington, J., Donnelly, D., Johnson, M.S., Sali, A. and Blundell, T.L.
Environment-specific amino acid substitution tables: tertiary templates and prediction of protein folds
Protein Science 1, 216-226 (1992)

LINK010101 Substitution matrices from an neural network model (Lin et al., 2001)
PMID:11694178
Lin, K., May, A.C. and Taylor, W.R.
Amino acid substitution matrices from an artificial neural network model
J Comput Biol. 8, 471-481 (2001)

BLAJ010101 Matrix built from structural superposition data for identifying potential remote homologues (Blake-Cohen, 2001)
PMID:11254392
Blake, J.D. and Cohen, F.E.
Pairwise sequence alignment below the twilight zone
J Mol Biol. 307, 721-735 (2001)

PRLA000101 Structure derived matrix (SDM) for alignment of distantly related sequences (Prlic et al., 2000)
PMID:10964983
Prlic, A., Domingues, F.S. and Sippl, M.J.
Structure-derived substitution matrices for alignment of distantly related sequences

Protein Eng. 13, 545-550 (2000)

PRLA000102 Homologous structure dereived matrix (HSDM) for alignment of distantly related sequences (Prlic et al., 2000)
PMID:10964983
Prlic, A., Domingues, F.S. and Sippl, M.J.
Structure-derived substitution matrices for alignment of distantly related sequences
Protein Eng. 13, 545-550 (2000)

DOSZ010101 Amino acid similarity matrix based on the sausage force field (Dosztanyi-Torda, 2001)
PMID:11524370
Dosztanyi, Z. and Torda, A.E.
Amino acid similarity matrices based on force fields
Bioinformatics. 17, 686-699 (2001) * #SM_SAUSAGE * #Amino acid similarity matrix based on the sausage force field * #Supplementary material * #http://www.rsc.anu.edu.au/~zsuzsa/suppl/matrices/SM_SAUSAGE * #Zsuzsanna Doszt?yi and Andrew E. Torda * #Amino acid similarity matrices based on force fields * #The amino acids are ordered according to the first principal component of the SM_SAUSAGE matrix. * #The native cysteine residues were devided into two subsets depending on their covalent state. * #Three rows correspond to cysteines: disulfide bonded (O), free cysteines (J) and all cysteines (C).

DOSZ010102 Normalised version of SM_SAUSAGE (Dosztanyi-Torda, 2001)
PMID:11524370
Dosztanyi, Z. and Torda, A.E.
Amino acid similarity matrices based on force fields
Bioinformatics. 17, 686-699 (2001) * #SM_SAUS_NORM * #Normalised version of SM_SAUSAGE * #For each matrix element of SM_SAUSAGE, the average over its column and row were subtracted. * #Supplementary material * #http://www.rsc.anu.edu.au/~zsuzsa/suppl/matrices/SM_SAUS_NORM * #Zsuzsanna Doszt?yi and Andrew E. Torda * #Amino acid similarity matrices based on force fields * #The amino acids are ordered according to the first principal component of the SM_SAUSAGE matrix.

DOSZ010103 An amino acid similarity matrix based on the THREADER force field (Dosztanyi-Torda, 2001)
PMID:11524370
Dosztanyi, Z. and Torda, A.E.
Amino acid similarity matrices based on force fields
Bioinformatics. 17, 686-699 (2001) * #SM_THREADER * #An amino acid similarity matrix based on the THREADER force field (Jones, DT et al.Nature, 358,86-89). * #Supplementary material * #http://www.rsc.anu.edu.au/~zsuzsa/suppl/matrices/SM_THREADER * #Zsuzsanna Doszt?yi and Andrew E. Torda * #Amino acid similarity matrices based on force fields * #The amino acids are ordered according to the first principal component of the SM_SAUSAGE matrix.

DOSZ010104 Normalised version of SM_THREADER (Dosztanyi-Torda, 2001)
PMID:11524370
Dosztanyi, Z. and Torda, A.E.
Amino acid similarity matrices based on force fields
Bioinformatics. 17, 686-699 (2001) * #SM_THREAD_NORM * #Normalised version of SM_THREADER * #based on the THREADER force field (Jones, DT et al.Nature, 358,86-89) * #For each matrix element of SM_THREADER, the average over its column and row were subtracted. * #Supplementary material * #http://www.rsc.anu.edu.au/~zsuzsa/suppl/matrices/SM_THREAD_NORM * #Zsuzsanna Doszt?yi and Andrew E. Torda * #Amino acid similarity matrices based on force fields * #The amino acids are ordered according to the first principal component of the SM_SAUSAGE matrix.

GIAG010101 Residue substitutions matrix from thermo/mesophilic to psychrophilic enzymes (Gianese et al., 2001)
PMID:11342709
Gianese, G., Argos, P. and Pascarella, S.
Structural adaptation of enzymes to low temperatures
Protein Eng. 14, 141-148 (2001) * (rows = WARM, cols = COLD)

DAYM780302 Log odds matrix for 40 PAMs (Dayhoff et al., 1978) R
Dayhoff, M.O., Schwartz, R.M. and Orcutt, B.C.
A model of evolutionary change in proteins
In "Atlas of Protein Sequence and Structure", Vol.5, Suppl.3 (Dayhoff, M.O., ed.), National Biomedical Research Foundation, Washington, D.C., p.352 (1978) * # * # This matrix was produced by "pam" Version 1.0.6 [28-Jul-93] * # * # PAM 40 substitution matrix, scale = ln(2)/2 = 0.346574 * # * # Expected score = -4.27, Entropy = 2.26 bits * # * # Lowest score = -15, Highest score = 13 * #

HENS920104 BLOSUM50 substitution matrix (Henikoff-Henikoff, 1992)
LIT:1902106 PMID:1438297
Henikoff, S. and Henikoff, J.G.
Amino acid substitution matrices from protein blocks
Proc. Natl. Acad. Sci. USA 89, 10915-10919 (1992) * # Matrix made by matblas from blosum50.iij * # BLOSUM Clustered Scoring Matrix in 1/3 Bit Units * # Blocks Database = /data/blocks_5.0/blocks.dat * # Cluster Percentage: >= 50 * # Entropy = 0.4808, Expected = -0.3573

QUIB020101 STROMA score matrix for the alignment of known distant homologs (Qian-Goldstein, 2002)
PMID:12211027
Qian, B. and Goldstein, R.A.
Optimization of a new score function for the generation of accurate alignments
Proteins. 48, 605-610 (2002)

VT160 T. Miller and M. Vingron Modeling Amino Acid Replacement Journal of Computational Biology, 7(6):761-776, 2000. Abstract: The estimation of amino acid replacement frequencies during molecular evolution is crucial for many applications in sequence analysis. Score matrices for database search programs or phylogenetic analysis rely on such models of protein evolution. Pioneering work was done by M. Dayhoff et al. (Atlas of Protein Sequences and Structure, 1978, 5, 345-352), who formulated a Markov model of evolution and derived the famous PAM score matrices. Her estimation procedure for amino acid exchange frequencies is restricted to pairs of proteins that have a constant and small degree of divergence. Here we present an improved estimator, called the resolvent method, that is not subject to these limitations. This extension of Dayhoff's approach enables us to estimate an amino acid substitution model from alignments of varying degree of divergence. Extensive simulations show the capability of the new estimator to recover accurately the exchange frequencies among amino acids. Based on the SYSTERS database of aligned protein families (Krause & Vingron, Bioinformatics, 1998, 14(5), 430-438) we recompute a series of score matrices.

Bacterial/Viruses Gene Finding

ABSplit

Program determines for the nucleotide sequence of approx. 300-600 n.p. whether it belongs to archeal or bacterial genome.

To classify the sequences linear discriminant analysis approach is used. Each sequence is represented by number of statistical parameters: mono- di- tri- nucleotide frequencies, and linear correlation coefficients (2 additional parameters) and mean absolute deviation (2 additional parameters) between the codon frequencies in the longest ORF found in the query sequence with the frequencies of codons in archaeal and bacterial genomes.

The training and testing data were taken from the sequences of the 157 genomes (21 archaeal and 136 bacterial). The length of sequences was 630. They were taken by splitting genomes to the sequences of this size, each 7-th fragment put in the testing set. There were 651612 fragments for training and 93008 fragments for testing data. The parameters for the linear discriminant function were obtained on the training set. The testing result in the following error estimates:

Number of sequences=93008 (class(A)=9158;class(B)=83850)
Archea(number/fraction)=18123/0.194854; mean_score=929428.413570
Bacteria(number/fraction)=74885/0.805146; mean_score=-1295582.386205
Test results:
Fraction of true predictions: 0.865141[80465]
Class 0: (Archea)
Fraction of true positives : 0.804652[7369]
Fraction of false negatives : 0.195348[1789]
Class 1: (Bacteria)
Fraction of true positives : 0.871747[73096]
Fraction of false negatives : 0.128253[10754]

The program has three output options:

- Output short statistics about the sequence set
- Write splitted sequence in two separate files (one file for predicted archeal and other for predicted bacterial sequences)
- Test output with prediction result for each sequence (if classification of sequences is established in FASAT file)

OUTPUT EXAMPLE

```
LDF discrimination threshold=0.000000
Prediction results:
Number of sequences=129
Arch(num/fract)=64/0.496124; mean_score=1173110.225735
Bact(num/fract)=65/0.503876; mean_score=-679245.160401

Histogram:
1       -1653112.270017-1492294.115256 0.007752
2       -1492294.115256-1331475.960496 0.015504
3       -1331475.960496-1170657.805735 0.015504
4       -1170657.805735-1009839.650974 0.038760
5       -1009839.650974  -849021.496214 0.069767
6        -849021.496214  -688203.341453 0.085271
```

```
7        -688203.341453 -527385.186693 0.093023
8        -527385.186693 -366567.031932 0.108527
9        -366567.031932 -205748.877172 0.023256
10       -205748.877172  -44930.722411 0.038760
11        -44930.722411  115887.432349 0.031008
12        115887.432349  276705.587110 0.054264
13        276705.587110  437523.741870 0.015504
14        437523.741870  598341.896631 0.023256
15        598341.896631  759160.051392 0.062016
16        759160.051392  919978.206152 0.023256
17        919978.206152 1080796.360913 0.015504
18       1080796.360913 1241614.515673 0.038760
19       1241614.515673 1402432.670434 0.046512
20       1402432.670434 1563266.457703 0.038760

Predicted archaeal sequences:
>AB001339|seq56|1
ttagtcagggggccccgccgatgaaaccggggacagctactaaacccattgccagtggtgg
tggtagctctggccctagtctgggctccggccaacccagagcagaacggcccggtggcggc
aatgcaggggcaaatgttggtcccattgcggccaatcccgttgctagtagtgctcccccta
aaccgaaaccaactcccagttcccccgctaagccagaccccttaaagtgcgttagccaatg
taaacccagttatccctccatcctccaggggggaagaaggtagtgctacagtattaatttca
gtaaatgatagtggtggtgtgaccagcgtaaccatcaccaatgcccacggcaacagcgagg
tcaaccgccaggccctattggcagccagaaaaatgcagtttacggcccccgccagtggtca
atccaaatcagtccctgtggtgattcacttcaccgttgctggttcagactttgatcgtcag
gcgagggagcgtcagcaacagcaggaagagttgcgtcaggccgcccgcagagcagaagagg
aaaaggcaaatcaagcccgtcagagacagttggaagaggagcgtcaagcccgccaacggca
attagagaaagaacgggaag
>AB001339|seq128|1
aggcttccaagcaagcttcaattaaggatttttccagaaagggatcccccacctgcaccgc
tgggcgatcgtccatggactgatccgttaactcagcactggcaaaactggctccccccatg
ccatcccgtcccgtggtggaaccgacatataaaactggattgcctatcccagaagccccag
ctttgacaatttcttccgtttccatcaaacccaaggccatggcgttgacgaggggattacc
ggagtaagccggatcaaagtagatttccccgcccacagtgggcacaccaacacaattaccg
taatgactgatcccatccactaccccggtgaaaatacgtcgattcctagcatcgtccaaat
taccgaaccgtagggaatttaaaatggcgatcggcctcgctcccatggtgaaaatatcccg
cagaatccccccctactccggtggcggctccctggaatggctccactgcggaaggatggtta
tgggattcgattttaaacgccaatctcaggccatcccccaaatctacgaccccggcatttt
ccccaggccccactaaaatgcgttctccttcggtgggaaagttactcagtaggggacggga
atttttataacaacaatgtt
>AB001339|seq184|1
attttcccgaagaaactacctccgatgcttggctgaccccagcagatgccggccaggatgg
tgatgcccaggaaccggcggaagatgggggagaagaaggagtagtgtcggaagaactggcc
ctgcctgaggacttacctcctatggatgccatggtggcggcagtggaagaaatgactccgg
tggtggtgcccgaaactgtaccagaaacagaaaccccagccttagaggatttggtcgccca
aaagaccgccctggaaaaggacattgccgctctgcaacgggaaaaagcccagtggtatggc
cagcagttccagcaattacagcgggaaatggcccggttagtggaggaaggcaccagggaat
tagggcaaagaaaagcagctctggaaaaggaaattgagaagttagagcgccgtcaggaacg
gattcaacaggaaatgcgtaccacttttgccggggcttcccaggagttggccatccgcgtg
cagggctttaaggattatttggtggggagtttgcaggatttggtttccgccgccgaccagt
tggaattaggggtgggggacagttgggagtcttcctctacccatggggatgcgattattga
aaatgccgacccaactccgg
>AB001339|seq336|1
tctgccagctttgccattaatttccgcctcgatcccaccgaggtcgttaccattcgccgca
cccaaggcacgttacaaaatattgtcgccaagattattgctccccaaacccaggaatcttt
taaaattgccgccgcgcgacgcacagtggaagaagccatcaccaaacggagcgagttgaag
gaagactttgataacgcccttaattcccgcctggagaaatacggcatcattgttctggaca
ccagtgtggtggatttagccttctcccccgaatttgccaaggcggtggaggaaaaacaaat
tgctgagcagagagcccagcgggcagtgtatgtggcccaggaagcggaacaacaggcccag
gcggacatcaaccgagccaaggggaaggcagaagcccaacggttactggcggaaactttaa
aagctcagggggggggaattagtcctacaaaaagaggcgatcgaagcttggcgggaaggggg
ggctcccatgcccaaggttttggtgatgggggagaaggcaaggggtctgcggttcccttt
atgtttaacctaactgacctggctaactagcggcagcggggaagttataggtcccagggct
```

```
cctgcctgacctttaggtcc
…
Predicted bacterial sequences:

>AB001339|seq8|1
ctgttacgtgttttgttgcaaacggaacttttttgcagtagttagctccgttgttgccgata
ccagtcaatggtattttcaatccttcccgcaagctcacctgggcttcaaacccaaattct
gctttagctttggtggtgtctaaacagcgacggggctggccgttgggttgatcggtttccc
aaataatgtccccctcaaactccatcagttcacagattaattccgttaagtctttgatgga
aatttcaaaattggtgcctaggttaaccggatcggctttgtcgtaggcttgggttcccatc
acaatgccccgggccgcatcagtggagtaaagaaattccctggtgggactgccgtcgcccc
aaacgggtaattgtttttgtccagcttttgcgcttcgtaaaccttatggatcaaggcagg
aatcacgtgggaactgcggggatcgaagttatcttctgggccgtaaagatttactggcaag
aggtaaatgccattaaagccatactgcaagcggtaggattccagttgcaccaacaatgctt
tcttggccacgccgtagggagcgttggtttcttcaggataaccgttccataagtcttcttc
cttaaagggtacaggggtaa
>AB001339|seq24|1
cctttttttatttatcttgcccgctcccaaattaaataatcaaacctaacgggtcaactcc
aaagacaacccaaggccattccaggctaattgattgaatcccgaattttattaactgtttg
ttccatttgtgccatgtttgcccctcgaccttggattgtggtccgtctccggtctttaccc
ctatcgtttcgcctcgatcgccatgtccccttggtaatgggattacttactgctctagcat
tattactatttattctcaatattagttgggggaatatcctgtccctcccttggcgatgct
ccaggccatctttgggctatctaccgatgctgaccatgaatttgtggtgcgtactctgcga
ttaccccggtccttggtggcattgttggtgggtatgggtttggcgatcgccggagggattt
tgcaaggcattacccgcaatcctttggcagcccctgaaattattggtgtcaatgcggggc
tagtttggtggcggttaccttcatcgttttgctaccgggtatttctccttccttgctgcca
gtggccgctttttgcggtggtttaacagcggcgatcgccatttatgtgctggcttggaatc
agggcagtgcccccgtccgg
>AB001339|seq32|1
atgatgttgattactcctccagtggcaccatccccgtaaatggccgttggcccctggatca
cttcaatccgttcaatggcactgggagcaatggtttgcaaatctcggaaggcattacggtt
ggtggtttggggcacaccgtcaatcaaaaccaaaacgttacgtcctcgcaaagcctggcca
aattgactggcactcccggtgctgggggctaagcctggcactagttgacccaaaatatccg
ccaaggaagagtaaacctgggtttgttgctcaatttctgcccgttcaattaccgttaccga
ccggggaatgttagcgatttcctcctctgtacgggtggcggaaaccacaatttgtagggcc
tcactttcctctatctcggcggttgtcccggcaacccctggtcgaatcagcaattgtaacc
cttgcgagttaggctttacttcggcttccggtggcccatttaccccgtgatagctaagcg
cacttggttatcggtcatttgggtaacactgacaaacgcaatgtccgcagtggggctcact
tcttcaaacccctggcccccaggtaaggccatcaaagtattgggaagatcaataattaagg
cattgcccaccgtttgtagg
>AB001339|seq64|1
ccgtccccgtcttaccggtaaagtatttgagaattagttgcagttaaggttgttcctcctg
tgttatcagatgccatggccggctgtctcaactaagaatttcaagctttggtgcaaggagt
gattatgaatcaagtacagtggtcggttttgttgatgggtatagtttcgctactatgtgct
cccagggcgtgggccgaaactaatccgaaccaattgaacaggacgaatattttagaatctg
gtaacttagaacgcaccaaagccggtgatttgctcccagttgcaaccactgttgatgagtg
gataacccaaattgcccaagcttcgatcatcgaaatcaaggaagcccggatcaatttgacc
gaagctggactggaactgaccctggctaccacgggccgcttatcaacaccaaccacttccg
tagtgggcaatgcactaattgtagatattcccaatgccatcctagccttgccggatagtga
cggactgcaacaggaaaaccccaccgaagaaattgccctagtgagcgttacagcattacct
gataatattgttcgcattgccattaccggggtcaatgtgccgccgacggttgaagttaatg
ccacagaccaatccctggta
…
```

ABSplit parameters:

Input	
Set of sequencese	Set of nucleotide sequences in 4-letter alphabet in FASTA format.
Output	
Discrimination data	Output file with discrimination result.
Format	Specifies output type: **Output short statistics**

	Write splitted sequences **Test output with prediction result**.
Archaea sequences	Output for predicted archaeal sequences.
Bacteria sequences	Output for predicted bacterial sequences.

BProm

BProm Prediction of bacterial promoters.

As a part of bacterial genome analysis suite of programs, and to enforce operon and gene prediction by FGENESB program, we introduce BProm, bacterial promoter prediction program.

Method description:

Algorithm predicts potential transcription start positions of bacterial genes regulated by sigma70 promoters (major E.coli promoter class). Linear discriminant function (LDF) combines characteristics describing functional motifs and oligonucleotide composition of these sites. BProm has accuracy of E.coli promoter recognition about 80%. Its specificity is also about 80% when tested on sets containing promoter and non-promoter sequences in equal numbers. It is not advisable to run BProm on whole genomes: To increase specificity, run BProm on a region between two neighboring ORFs located on the same strand, or on a sequence upstream from an ORF, keeping in mind that most promoters are located within 150 bp region from protein coding sequence.

BProm output:

First line - name of your sequence;

Second and Third lines - LDF threshold and the length of presented sequence

4th line - The number of predicted promoters

Next lines - positions of predicted promoters, and their scores with 'weights' of two conserved promoter boxes. Promoter position assign to the first nucleotide of the transcript (Transcription Start Site position).

After that we present elements of Transcriptional factor binding sites for each predicted promoter (if they found).

For example:

```
BProm  Sat Jan 18 21:11:25 EST 2003
>Region    of    E.coli    genome    between    protein_id="AAC76687.1"    and
protein_id="AAC7668
 Length of sequence-       420
 Threshold for promoters -  0.20
 Number of predicted promoters -      1
 Promoter Pos:    145 LDF-  6.02
 -10 box at pos.    130 ctttatgat Score    66
 -35 box at pos.    109 tttaat    Score    36

 Oligonucleotides from known TF binding sites:

 For promoter at    145:
        fis:  TCTTTAAT at position     107 Score -   6
     rpoD17:  TTATGATA at position     132 Score -   7
       lexA:  ATAAATAA at position     137 Score -  14
     rpoD17:  ATAATAAT at position     141 Score -   8
```

Parameters:

Input	
Sequences set	Input file.
Output	
Result	Name of the output file

FgenesB

Bacterial Operon and Gene Prediction.

FgenesB - Suite of Bacterial Operon and Gene Finding Programs

FgenesB is the most accurate *ab initio* prokaryotic gene prediction engine (see Table 1 at the bottom for its comparison with two other popular gene prediction programs). FgenesB gene prediction algorithm is based on Markov chain models of coding regions and translation and termination sites. The program uses genome-specific parameters learned by FGENESB-train script, which requires only DNA sequence from genome of interest as an input. (If you need parameters for your new bacteria, please contact Softberry.) FgenesB also includes simplified prediction of operons based only on distances between predicted genes.

FgenesB is gene finding part of **FgenesB_Annotator** which is a package for automatic annotation of bacterial genomes and includes the following features:

- automatic training of gene finding parameters for new bacterial genomes using only genomic DNA as an input (optionally, pre-learned parameters from related organism can be used);
- mapping of tRNA and rRNA genes;
- highly accurate Markov chains-based gene prediction;
- prediction of promoters and terminators;
- operon prediction based on distances between ORFs and frequencies of different genes neighboring each other in known bacterial genomes, as well as on promoter and terminator predictions;
- automatic annotation of predicted genes by homology with protein (COG, NR) databases.

For community sequence annotation, **ABsplit** (www.softberry.com/berry.phtml?topic=absplit&group=programs&subgroup=gfindb) program can be used that separates archaebacterial and eubacterial sequences.

FgenesB was used in first ever published bacterial community annotation project: see Tyson *et al.*, (2004) *Nature* 428(6978), 37-43.

Example of FgenesB output:

```
1      1 Op  1  21/0.000   +     CDS       407 -       1747   1311
2      1 Op  2   3/0.019   +     CDS      1926 -       3065   1237
3      2 Op  1   4/0.002   +     CDS      3193 -       3405    278
4      2 Op  2   4/0.002   +     CDS      3418 -       4545    899
5      2 Op  3  16/0.000   +     CDS      4578 -       6506   2148
6      2 Op  4      .      +     CDS      6595 -       9066   2957
7      3 Op  1      .      -     CDS     14175 -      14363    158
8      3 Op  2      .      -     CDS     14353 -      15249    351
9      3 Op  3      .      -     CDS     15170 -      15352     99
```

Table 1. Accuracy of prediction estimated on B.subtilis sequence: Frequency of genes starting from start codon other than first - 19.1% Borodovsky et al. (see GeneMark WEB pages (opal.biology.gatech.edu/GeneMark/genemarks.cgi)) has calculated accuracy for all genes, and has constructed three sets of difficult short genes (L ? 300bp) that have protein similarity support. There genes were used to demonstrate that short genes also can be predicted reasonably

well. First set (51set) has 51 genes with at least 10 strong similarities to known proteins. Then, 72set has 72 genes with at least two strong similarities, and 123set has 123 genes with at least one protein homolog.

Here are the prediction results on these three sets for GeneMarkS and Glimmer (calculated in Nucleic Acids Research, 2001, Vol. 29, No. 12, 2607-2618.) and FgenesB (calculated by Softberry, three iterations of FgenesB-train script):

```
               Sn (exact        Sn (exact+overlapping
                  predictions)      predictions)



123set:
Glimmer          57.0%           91.1
GeneMarkS        82.9            91.9
FgenesB          89.3            98.4


72set:
Glimmer          57.0%           91.7
GeneMarkS        88.9            94.4
FgenesB          91.5            98.6


51set:
Glimmer          51.0%           88.2
GeneMarkS        90.2            94.1
FgenesB          92.0            98.0


All genes of B.subtilis genome(GenBabk annotation):

Glimmer          62.4%           98.1
GeneMarkS        83.2            96.7
FgenesB          83.8            98.7
```

Please note that many genes in GenBank were annotated using GeneMark program, which should result in overestimation of its accuracy

Parameters:

Input	
Sequences	Browse your source file with nucleotide sequences in FASTA format.
Output	
Result	Name of the output file with prediction results.
Options	
Organism	Select parameter file for specified organizm.
Translation tabler	Select translation table (**Bacterial** is default): **Standart (1)** **Vertebrate Mitochondrial (2)** **Yeast Mitochondrial (3)** **Protozoan Mitochondrial and other (4)** **Invertebrate Mitochondrial (5)** **Ciliate Nuclear and other (6)** **Echinodermata Nuclear (9)** **Euplotid Nuclear (10)** **Bacterial (11)**

	Alternative Yeast Nuclear (12) **Ascidian Mitochondrial (13)** **Flatworm Mitochondrial (14)** **Blepharisma Macronuclear (15)**

FgenesB-Annotator

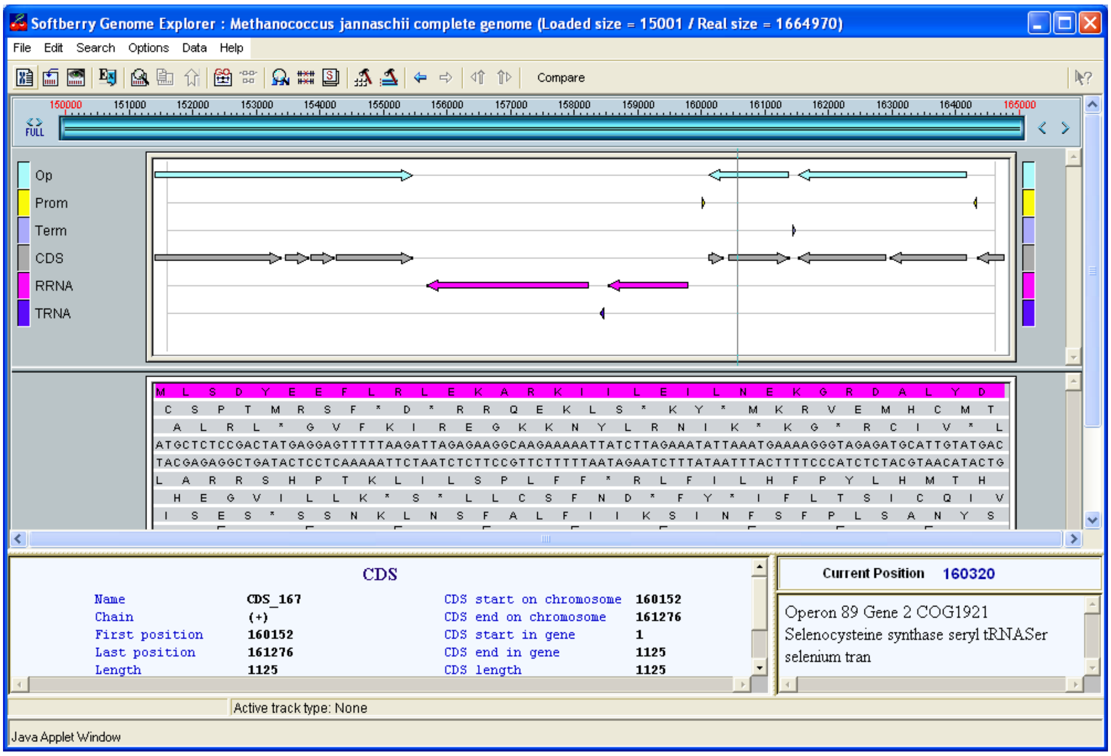
To identify protein and RNA genes in bacterial genomic sequences or environmental samples, Softberry developed Fgenesb_annotator pipeline that provides completely automatic, comprehensive annotation of bacterial sequences. The pipeline includes protein, tRNA and rRNA genes identification, finds potential promoters, terminators and operon units.

Predicted genes are annotated based on comparison with known proteins. The package provides options to work with a set of sequences such as scaffolds of bacterial genomes or short reads of DNA extracted from a bacterial community. The final annotation can be presented in GenBank form to be readable by visualization software such as Artemis [1] and GenomeExplorer (fig. 1 and 2). The gene prediction algorithm is based on Markov chain models of coding regions and translation and termination sites. For annotation of mixed bacterial community, we use special parameters of gene prediction computed based on a large set of known bacterial sequences. Operon models are based on distances between ORFs, frequencies of different genes neighboring each other in known bacterial genomes, and information from predicted potential promoters and terminators. The parameters of gene prediction are automatically trained during initial steps of sequence analysis, so the only input necessary for annotation of a new genome is its sequence. Optionally, parameters from closely related genomes can be used, instead of training new parameters. Bacterial gene/operon prediction and annotation requires, besides Fgenesb_annotator programs and scripts, BLAST, NCBI Non-Redundant database (NR), and a file reconstructed from COG database [2]. RRNA genes are annotated using BLAST similarity with all known bacterial rRNA database. For prediction of tRNA genes, the pipeline uses tRNAscan-SE package [3].

1. K. Rutherford, J. Parkhill, J. Crook, T. Horsnell, P. Rice, M-A. Rajandream and B. Barrell (2000) Artemis: sequence visualisation and annotation. Bioinformatics 16 (10) 944-945.
2. Tatusov RL, Natale DA, Garkavtsev IV, Tatusova TA, Shankavaram UT, Rao BS, Kiryutin B, Galperin MY, Fedorova ND, Koonin EV. (2001) The COG database: new developments in phylogenetic classification of proteins from complete genomes. Nucleic Acids Res. 29, 22-28.
3. Lowe, T.M. & Eddy, S.R. (1997) "tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence", Nucl. Acids Res., 25, 955-964.

The main features of Fgenesb_annotator are:

- Automatic training of gene finding parameters for new bacterial genomes using only genomic DNA as an input
- Optionally, pre-learned parameters from related organism can be used
- Optionally, generic Bacterial, Archaebacterial, or combined parameters can be used
- Mapping of tRNA and rRNA genes
- Highly accurate Markov chains-based gene prediction
- Prediction of promoters and terminators
- Operon prediction based on distances between ORFs and frequencies of different genes neighboring each other in known bacterial genomes, as well as on promoter and terminator predictions
- Automatic annotation of predicted genes by homology with COG, KEGG and NR databases.



```
M  L  S  D  Y  E  E  F  L  R  L  E  K  A  R  K  I  I  L  E  I  L  N  E  K  G  R  D  A  L  Y  D
 C  S  P  T  M  R  S  F  *  D  *  R  R  Q  E  K  L  S  *  K  Y  *  M  K  R  V  E  M  H  C  M  T
  A  L  R  L  *  G  V  F  K  I  R  E  G  K  K  N  Y  L  R  N  I  K  *  K  G  *  R  C  I  V  *  L
ATGCTCTCCGACTATGAGGAGTTTTTAAGATTAGAGAAGGCAAGAAAAATTATCTTAGAAATATTAAATGAAAAGGGTAGAGATGCATTGTATGAC
TACGAGAGGCTGATACTCCTCAAAAATTCTAATCTCTTCCGTTCTTTTTAATAGAATCTTTATAATTTACTTTTCCCATCTCTACGTAACATACTG
L  A  R  R  S  H  P  T  K  L  I  L  S  P  L  F  F  *  R  L  F  I  L  H  F  P  Y  L  H  M  T  H
  H  E  G  V  I  L  L  K  *  S  *  L  L  C  S  F  N  D  *  F  Y  *  I  F  L  T  S  I  C  Q  I  V
 I  S  E  S  *  S  S  N  K  L  N  S  F  A  L  F  I  I  K  S  I  N  F  S  F  P  L  S  A  N  Y  S
```

CDS

Name	CDS_167	CDS start on chromosome	160152
Chain	(+)	CDS end on chromosome	161276
First position	160152	CDS start in gene	1
Last position	161276	CDS end in gene	1125
Length	1125	CDS length	1125

Current Position 160320

Operon 89 Gene 2 COG1921
Selenocysteine synthase seryl tRNASer
selenium tran

Active track type: None

Java Applet Window

Fig.1. Bacterial Genome Explorer to work with annotations and comparison of genomes.

The package includes options to work with a set of sequences such as scaffolds of bacterial genomes, or short sequencing reads extracted from bacterial communities. For community sequence annotation, we developed ABsplit program that separates archaebacterial and eubacterial sequences (available separately). Final annotation can be presented in GenBank format to be readable by visualization software such as Artemis or Softberry Bacterial Genome Explorer (fig. 1 and 2, GenBank parser is available separately).

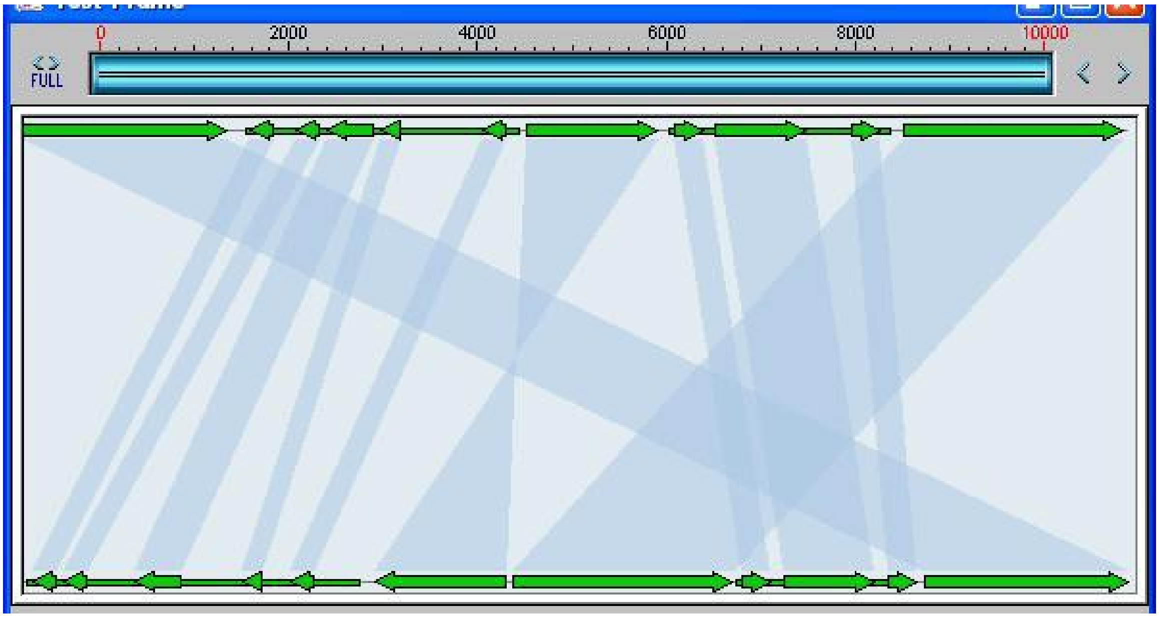


Fig.2. Comparison of two bacterial genomes view of Genome Explorer.

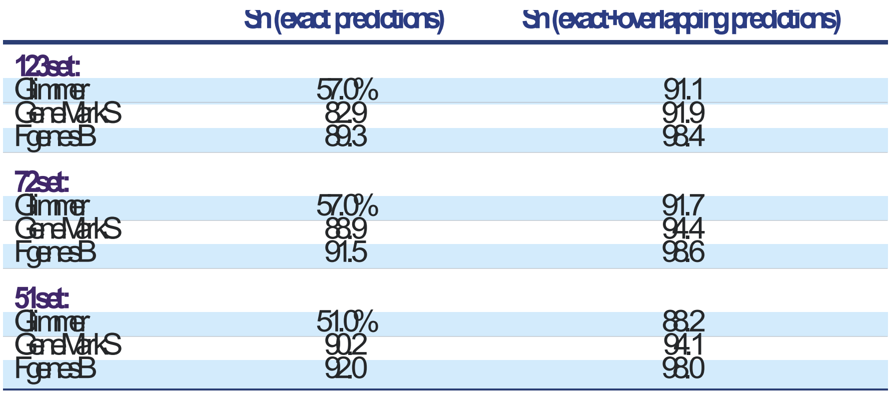
Main Steps of FGENESB annotation.

Many steps are optional and can be switched ON/OFF in configuration file.

STEP 1. Finds all potential ribosomal RNA genes using BLAST against bacterial and/or archaeal rRNA databases, and masks detected rRNA genes.
STEP 2. Predicts tRNA genes using tRNAscan-SE program (Washington University) and masks detected tRNA genes.
STEP 3. Initial predictions of long ORFs that are used as a starting point for calculating parameters for gene prediction. Iterates until stabilizes. Generates parameters such as 5th-order in-frame Markov chains for coding regions, 2nd-order Markov models for region around start codon and upstream RBS site, stop codon and probability distributions of ORF lengths.
STEP 4. Predicts operons based only on distances between predicted genes.
STEP 5. Runs BLAST for predicted proteins against COG database, cog.pro.
STEP 6. Finds conserved operonic pairs from blast output through cog data.
STEP 7. Uses information about conservation of neighboring gene pairs in known genomes to improve operon prediction.
STEP 8. Runs BLAST for predicted proteins against KEGG database.
STEP 9. Runs BLAST for predicted proteins against NR database.
STEP 10. Adds names of homologs from COG/KEGG/NR (found through BLAST) to annotation file (file with prediction results).
STEP 11. Predicts potential promoters (BPROM program) or terminators (BTERM) in upstream and downstream regions, correspondingly, of predicted genes. BTERM is the program predicting bacterial-independent terminators with energy scoring based on discriminant function of hairpin elements.
STEP 12. Refines operon predictions using predicted promoters and terminators as additional evidences.

FGENESB gene prediction engine is one of the most accurate prokaryotic gene finders available: see Table 1 for its comparison with two other popular gene prediction programs.

Table 1. Comparison of three popular bacterial gene finders. Accuracy estimate was done on a set of difficult short genes that was previously used for evaluating other bacterial gene finders (http://opal.biology.gatech.edu/GeneMark/genemarks.cgi). First set (51set) has 51 genes with at least 10 strong similarities to known proteins. Then 72set has 72 genes with at least two strong similarities, and 123set has 123 genes with at least one protein homolog.
Here are the prediction results on these three sets for GeneMarkS and Glimmer (calculated by Besemer et al. (2001) Nucl. Acids Res. 29:2607-2618) and FGENESB gene prediction engine (calculated by Softberry).



	Sn (exact predictions)	Sn (exact+overlapping predictions)
123set:		
Glimmer	57.0%	91.1
GeneMarkS	82.9	91.9
FgenesB	89.3	98.4
72set:		
Glimmer	57.0%	91.7
GeneMarkS	88.9	94.4
FgenesB	91.5	98.6
51set:		
Glimmer	51.0%	88.2
GeneMarkS	90.2	94.1
FgenesB	92.0	98.0

All prediction components of FGENESB are extremely fast (minutes per genome). The limiting stage is BLAST annotation, which for *E.coli* genome takes around 12 hours on a single processor. Using multiple processors and corresponding BLAST would speed up annotation proportionally.

Explanation of Fgenesb_annotator output

Example of FGENESB output:

```
Prediction of potential genes in microbial  genomes
 Time:   Tue Aug 22 11:21:15 2006
 Seq name: gi|15807672|ref|NC_001264.1| Deinococcus radiodurans R1 (partial sequence)
 Length of sequence - 54865 bp
 Number of predicted genes - 48, with homology - 48
 Number of transcription units - 18, operons - 13 average op.length -  3.3

     N      Tu/Op   Conserved  S             Start         End    Score
                   pairs(N/Pv)

                                -   TRNA        147 -       222   78.9  # Arg CCG 0 0
                                +   TRNA        315 -       398   63.6  # Leu TAG 0 0
                                + 5S_RRNA       521 -       637  100.0  # AB001721 [D:2735..2851]
                                + SSU_RRNA      698 -      2181  100.0  # SSU_RRNA ##
                                + LSU_RRNA     2302 -      5345  100.0  # BX248583 [R:613128..616171]
                                +    Prom      5304 -      5363   41.4
 1     1 Op  1  22/0.000    +    CDS       5410 -      6300    498  ## COG1192 ATPases involved …
 2     1 Op  2     .        +    CDS       6297 -      7178    502  ## COG1475 Predicted …
                                +    Term      7203 -      7253    9.1
                                -    Term      7191 -      7241   14.2
 3     2 Tu  1     .        -    CDS       7283 -      8746    909  ## COG1012 NAD-dependent …
                                -    Prom      8792 -      8851    2.8
 4     3 Tu  1     .        +    CDS       8802 -      9533    302  ## COG2068 Uncharacterized …
                                +    Term      9779 -      9818    3.8
                                -    Term      9527 -      9567    9.0
 5     4 Op  1   2/0.125    -    CDS       9584 -     10762   1005  ## COG1063 Threonine …
 6     4 Op  2     .         -    CDS      10759 -     11457    666  ## COG5637 Predicted integral
…
                                -    Prom     11697 -     11756    2.4
 7     5 Op  1  37/0.000    +    CDS      11704 -     12609    872  ## COG1131 ABC-type multidrug
…
 8     5 Op  2   5/0.000    +    CDS      12726 -     13517    812  ## COG0842 ABC-type multidrug
…
 9     5 Op  3  15/0.000    +    CDS      13674 -     14684   1028  ## COG4585 Signal transduction
…
10     5 Op  4     .         +    CDS      14681 -     15316    506  ## COG2197 Response regulator
…
…
47    18 Op  1     .        -    CDS      53783 -     54703    431  ## DRA0045 hypothetical …
48    18 Op  2     .        -    CDS      54700 -     54864     91  ## DRA0046 hypothetical …

Predicted protein(s)

>gi|15807672|ref|NC_001264.1| GENE      1       5410   -       6300     498     296 aa, chain + ##
HITS:3   COG:DRA0001  KEGG:FRAAL2247  NR:6460595  ##  COG:  DRA0001  COG1192  #  Protein_GI_number:
15807673 # Func_class: D Cell cycle control, cell division, chromosome partitioning  # Function:
ATPases involved in chromosome partitioning # Organism: Deinococcus radiodurans # 37      296  1
260      260      459  100.0  1e-129 ## KEGG: FRAAL2247 # Name: not_defined # Def: chromosome
partitioning protein (partial match) [EC:2.7.10.2] # Organism: F.alni # Pathway: not_defined # 48
283      50      291      302      118    35.0  5e-26 ## NR: gi|6460595|gb|AAF12301.1| chromosome
partitioning  ATPase,  putative,  ParA  family  [Deinococcus  radiodurans  R1]^Agi|15807673|ref|
NP_285325.1| chromosome partitioning ATPase, putative, ParA family [Deinococcus radiodurans R1] #
37     296  1     260     260     459  100.0  1e-128
VLKNHLFLRNLIFSVLPVVQHFLTFKEEQSIADLSDMVSAVKTLTVFNHAGGAGKTSLTL
NVGYELARGGLRVLLLDLDPQANLTGWLGISGVTREMTVYPVAVDGQPLPSPVKAFGLDV
IPAHVSLAVAEGQMMGRVGAQGRLRRALAEVSGDYDVALIDSPPSLGQLAILAALAADQM
IVPVPTRQKGLDALPGLQGALTEYREVRPDLTVALYVPTFYDARRRHDQEVLADLKAHLS
PLARPVPQREAVWLDSTAQGAPVSEYAPGTPVHADVQRLTADIAAAIGVAYPGENA

>gi|15807672|ref|NC_001264.1| GENE      2       6297   -       7178     502     293 aa, chain + ##
HITS:3   COG:DRA0002  KEGG:SAR11_0354  NR:12230476  ##  COG:  DRA0002  COG1475  #  Protein_GI_number:
15807674  #  Func_class:  K  Transcription   #  Function:  Predicted  transcriptional  regulators  #
Organism: Deinococcus radiodurans # 1     293       1     293     293      478  100.0  1e-135 ##
KEGG:
 SAR11_0354 # Name: parB # Def: chromosome partitioning protein [EC:2.7.7.-] # Organism: P.ubique
# Pathway: not_defined # 10    200      12     177     282     107   36.0  7e-23 ## NR: gi|
12230476|sp|Q9RZE7|PARB2_DEIRA  Probable  chromosome  2  partitioning  protein  parB  (Probable
chromosome  II  partitioning  protein  parB)^Agi|6460594|gb|AAF12300.1|  chromosome  partitioning
protein,  ParB  family  [Deinococcus  radiodurans  R1]^Agi|15807674|ref|NP_285326.1|  chromosome
partitioning protein, ParB family [Deinococcus radiodurans R1] # 1     293        1     293
293    478  100.0  1e-133
MTRRRPERRRDLLGLLGETPVDLSQANDIRALPVNELKVGSTQPRRSFDLERLSELAESI
RAHGVLQPLLVRSVDGQYEIVAGERRWRAAQLAGLAEVPVVVRQLSNEQARAAALIENLQ
RDNLNVIDEVDGKLELIALTLGLEREEARKRLMQLLRAVPGDEHEQLDQVFRSMGETWRT
FAKNKLRILNWPQPVLEALRAGLPLTLGSVVASAPPERQAELLKLAQNGASRSQLLQALQ
TPSQTSAVTPEHFAKVLSSKRFLSGLDTPTREALDRWLARMPERVRQAIDEQS

...
```

Example of FGENESB output in GenBank format (scripts run_tgb.pl, togenbank.pl):

```
     gene            complement(147..222)
                     /gene="Arg CCG"
     tRNA            complement(147..222)
                     /gene="Arg CCG"
                     /product="tRNA-Arg"
                     /note="Arg CCG 0 0"
     gene            315..398
                     /gene="Leu TAG"
     tRNA            315..398
                     /gene="Leu TAG"
                     /product="tRNA-Leu"
                     /note="Leu TAG 0 0"
     gene            521..637
                     /gene="AB001721 [D:2735..2851]"
     rRNA            521..637
                     /gene="AB001721 [D:2735..2851]"
                     /product="5S ribosomal RNA"
                     /note="AB001721 [D:2735..2851]"
     gene            698..2181
                     /gene="SSU_RRNA"
     rRNA            698..2181
                     /gene="SSU_RRNA"
                     /product="16S ribosomal RNA"
                     /note="SSU_RRNA"
     gene            2302..5345
                     /gene="BX248583 [R:613128..616171]"
     rRNA            2302..5345
                     /gene="BX248583 [R:613128..616171]"
                     /product="23S ribosomal RNA"
                     /note="BX248583 [R:613128..616171]"
     promoter        5304..5363
     CDS             5410..6300
                     /function="ATPases involved in chromosome partitioning"
                     /note="Operon 1 Gene 1 COG1192 ATPases involved in
                     chromosome partitioning"
                     /translation="VLKNHLFLRNLIFSVLPVVQHFLTFKEEQSIADLSDMVSAVKTL
                     TVFNHAGGAGKTSLTLNVGYELARGGLRVLLLDLDPQANLTGWLGISGVTREMTVYPV
                     AVDGQPLPSPVKAFGLDVIPAHVSLAVAEGQMMGRVGAQGRLRRALAEVSGDYDVALI
                     DSPPSLGQLAILAALAADQMIVPVPTRQKGLDALPGLQGALTEYREVRPDLTVALYVP
                     TFYDARRRHDQEVLADLKAHLSPLARPVPQREAVWLDSTAQGAPVSEYAPGTPVHADV
                     QRLTADIAAAIGVAYPGENA"
                     /transl_table=11
     CDS             6297..7178
                     /function="Predicted transcriptional regulators"
                     /note="Operon 1 Gene 2 COG1475 Predicted transcriptional
                     regulators"
                     /translation="MTRRRPERRRDLLGLLGETPVDLSQANDIRALPVNELKVGSTQP
                     RRSFDLERLSELAESIRAHGVLQPLLVRSVDGQYEIVAGERRWRAAQLAGLAEVPVVV
                     RQLSNEQARAAALIENLQRDNLNVIDEVDGKLELIALTLGLEREEARKRLMQLLRAVP
                     GDEHEQLDQVFRSMGETWRTFAKNKLRILNWPQPVLEALRAGLPLTLGSVVASAPPER
                     QAELLKLAQNGASRSQLLQALQTPSQTSAVTPEHFAKVLSSKRFLSGLDTPTREALDR
                     WLARMPERVRQAIDEQS"
                     /transl_table=11
     terminator      7203..7253
     terminator      complement(7191..7241)
     CDS             complement(7283..8746)
                     /function="NAD-dependent aldehyde dehydrogenases"
                     /note="Operon 2 Gene 1 COG1012 NAD-dependent aldehyde
                     dehydrogenases"
                     /translation="MTTTDLRTTYSSVTRSQAYFDGEWRNAPRNFEVRHPGNGEVIGE
                     VADCTPTDARQAIDAAEVALREWRQVNPYERGKILRRWHDLMFEHKEELAQLMTLEMG
                     KPISETRGEVHYAASFIEWCAEEAGRIAGERINLRFPHKRGLTISEPVGIVYAVTPWN
                     FPAGMITRKAAPALAAGCVMILKPAELSPMTALYLTELWLKAGGPANTFQVLPTNDAS
                     ALTQPFMNDSRVRKLTFTGSTEVGRLLYQQAAGTIKRVSLELGGHAPFLVFDDADLER
                     AASEVVASKFRNSGQTCVCTNRVYVQRGVAEEFIRLLTEKTAALQLGDPFDEATQVGP
                     VVEQAGLDKVQRQVQDALTKGAQATTGGQVSSGLFFQPTVLVDVAPDSLILREETFGP
                     VAPVTIFDTEEEGLRLANDSEYGLAAYAYTRDLGRAFRIAEGLEYGIVGINDGLPSSA
                     APHVPFGGMKNSGVGREGGHWGLEEYLETKFVSLGLS"
                     /transl_table=11
     promoter        complement(8792..8851)

...

BASE COUNT     11009 a   16099 c   16880 g   10877 t
ORIGIN
        1 tctttgctcg ccatacccaa agtctacacg ctgattttca cgtttccaga ccctgccctc
       61 tcgctactca gctctccaag tttgctcgct tgatgaatga tcaaatcttt taaagataaa
      121 agccatgcgt gaggctagat caacccttgt gcccccggca ggattcgaac ctgcggcctt
...
    54841 gtcgcccagt tgaatggctc gccac
//
```

Example of FGENESB output in Sequin format:

```
>Feature test_seq
222     147     gene
                        locus_tag       C8J_0001
222     147     tRNA
                        product tRNA-Arg
                        inference       profile:tRNAscan-SE:1.23
315     398     gene
                        locus_tag       C8J_0002
315     398     tRNA
                        product tRNA-Leu
                        inference       profile:tRNAscan-SE:1.23
521     637     gene
                        locus_tag       C8J_0003
521     637     rRNA
                        product 5S ribosomal RNA
698     2181    gene
                        locus_tag       C8J_0004
698     2181    rRNA
                        product 16S ribosomal RNA
2302    5345    gene
                        locus_tag       C8J_0005
2302    5345    rRNA
                        product 23S ribosomal RNA
5304    6300    gene
                        locus_tag       C8J_0006
5304    5363    promoter
5410    6300    CDS
                        product hypothetical protein
                             note     similar to D.radiodurans chromosome partitioning
ATPase …
                        protein_id      gnl|bbsrc|C8J_0006
                        inference       ab initio prediction:Fgenesb:2.0
6297    7253    gene
                        locus_tag       C8J_0007
6297    7178    CDS
                        product chromosome partitioning protein, ParB family
                        protein_id      gnl|bbsrc|C8J_0007
                        inference       ab initio prediction:Fgenesb:2.0
7203    7253    terminator
```

```
8851    7191    gene
                        locus_tag       C8J_0008
7241    7191    terminator
8746    7283    CDS
                        product succinate-semialdehyde dehydrogenase
                        EC_number       1.2.1.16
                        protein_id      gnl|bbsrc|C8J_0008
                        inference       ab initio prediction:Fgenesb:2.0
8851    8792    promoter
...
```

Description of Fgenesb_annotator output fields:

For each genomic sequence (complete genome, scaffold, read, etc.) the program lists locations of predicted ORFs, rRNAs, tRNAs, promoters and terminators.

ORFs are labeled as CDS and provided with their order number in a sequence and an indicator of whether they are transcribed as a single transcription unit (Tu) or in operons (Op) (of course these are predictions).
If an ORF has a homolog, its short name is provided after a "##" separator (here name of only one homolog - either from COG, KEGG, or NR - is given; best homologs from all databases are listed in ID lines of predicted proteins, see below).

For example:

```
  5   4 Op   2   +   CDS 2737 - 3744   871   ## COG0673 Predicted dehydrogenases
```

is description for predicted gene number 5 in 4th Operon with coordinates 2737 - 3744 in the '+' strand and it is the second gene in operon.
Coding chain for this CDS (+) means a direct chain, (-) means a complementary chain.
871 is a score of gene homology assigned by BLAST, and COG0673 is an ID of its homolog from the COG database.

In other words, first column lists an ordered number of predicted CDS, starting from beginning of a sequence; second column – number of predicted operon/TU, and fourth column – number of gene in an operon (always 1 for a TU).

For some operons, we report supportive evidence related to conservation in relative locations of genes in predicted operon in different bacteria. For example:

```
3      2 Op   1    4/0.002    +     CDS        3193 -       3405     278   ## COG2501
Uncharacterized ACR
```

Here, in 4/0.002, 4 is a number of observations of this gene being next to one of its neighbors on known bacterial genomes (we call it N-value), while 0.002 is a P-value, an empirical probability of observing N occurrences of genes being adjacent by random chance. P is a very approximate measure. For all P<0.0001, the value in output is 0.000.

At the end of annotation, we also provide protein products of predicted genes in fasta format, with full name of homolog and homology scores according to BLAST.

Information about homologs is given in ID lines of predicted proteins, for example:

```
>gi|15807672|ref|NC_001264.1|  GENE      7      11704   -      12609     872     301 aa,
chain  +  ##  HITS:3   COG:DRA0007  KEGG:DRA0007  NR:6460585  ##  COG:  DRA0007  COG1131  #
```

```
Protein_GI_number: 15807679 # Func_class: V Defense mechanisms  # Function: ABC-type
mult
idrug transport system, ATPase component # Organism: Deinococcus radiodurans # 1
301      1      301      301      503  100.0  1e-142 ## KEGG: DRA0007 # Name:
not_defined # Def: putative ABC-2 type transport system ATP-binding protein #
Organism: D.radiodurans # Pathway: ABC transporters - General [PATH:dra02010] # 1
301      1     301     301     503  100.0  1e-142 ## NR: gi|6460585|gb|AAF12291.1|
ABC transporter, ATP-binding protein, putative [Deinococcus radiodurans R1]^Agi|
15807679|ref|NP_285331.1| ABC transporter, ATP-binding protein, putative [Deinococcus
radiodurans R1] # 1    301      1    301    301 503  100.0  1e-141
MITTFEQVSKTYGHVTALSDFNLTLRTGELTALLGPNGAGKSTAIGLLLGLSAPSAGQVR
VLGADPRRNDVRARIGAMPQESALPAGLTVREAVTLFASFYPAPLGVDEALALADLGPVA
GRRAAQLSGGQKRRLAFALAVVGDPELLLIDEPTTGMDAQSRAAFWEAVTGLRARGRTIL
LTTHYLEEAERTADRVVVMNGGRILADDTPQGLRSGVGGARVSFVSDLVQAELERLPGVS
AVQVDAAGRADLRTSVPEALLAALIGSGTTFSDLEVRRATLEEAYLQLTGPQDMTAVTRS
A
```

While looking a bit complex for a human eye, it is well suited for parsing by a program.

ID lines of predicted proteins consist of the following parts that are separated from each other by “##” separator:

```
>gi|15807672|ref|NC_001264.1| GENE     7     11704   -     12609    872    301 aa,
chain +
```

(sequence name, gene number, coordinates of a gene, length of a corresponding protein, chain)

```
## HITS:3  COG:DRA0007 KEGG:DRA0007 NR:6460585
```

(shows the number of homologs found in protein databases (takes into account maximum one best homolog per a database), lists homologs IDs in the format DB:ID (e.g., COG:DRA0007); notes:

- for homologs from NR, gi- numbers are given as homologs IDs;
- DB:ns indicates that a protein DB was not searched (e.g., NR:ns);
- DB:no indicates that a protein DB was searched but no homologs were found (e.g., NR:no))

Then, complete ID lines of homologs are given preceded by DB names where they were found by BLAST (e.g., NR:) and followed by statistics from corresponding BLAST outputs.

```
## COG: DRA0007 COG1131 # Protein_GI_number: 15807679 # Func_class: V Defense
mechanisms  # Function: ABC-type multidrug transport system, ATPase component #
Organism: Deinococcus radiodurans # 1    301      1    301    301    503  100.0
1e-142

## KEGG: DRA0007 # Name: not_defined # Def: putative ABC-2 type transport system ATP-
binding protein # Organism: D.radiodurans # Pathway: ABC transporters - General
[PATH:dra02010] # 1    301      1    301    301    503  100.0  1e-142

## NR: gi|6460585|gb|AAF12291.1| ABC transporter, ATP-binding protein, putative
[Deinococcus radiodurans R1]^Agi|15807679|ref|NP_285331.1| ABC transporter, ATP-
binding protein, putative [Deinococcus radiodurans R1] # 1     301      1     301
301 503  100.0  1e-141
```

BLAST parameters of similarity found for predicted protein are shown in the following order:
Start and stop of region of similarity (1 301) in predicted protein
Start and stop of region of similarity (1 301) in homolog from a database
Length of homologous protein (301)
BLAST score (503) and Identity (100.0 %)
BLAST Expected value (1e-141)

For other predictions (rRNA, promoters, etc.) we provide only description lines, for example:

```
      - LSU_RRNA   884415 -    887254   98.0  # Leuconostoc oenos S60377
```

rRNAs are labeled as LSU_RRNA, SSU_RRNA or 5S_RRNA (large subunit, small subunit, and 5S), tRNAs as TRNA, promoters as Prom, and terminators as Term.

Terminator regions (their coordinates and scores) are reported by FindTerm program:

```
      +    Term        492 -        537   -0.9
```

Promoters (their coordinates and scores) are reported by BPROM program.

Parameters:

	Input
Sequences	Name of the input file with sequences in FASTA format (4-letters alphabet).
	Output
Prediction result	Name of the output file with prediction results.
Genbank output	Name of the output file in Genbank format.
	Options
Base	Gene finding parameters used for initial gene prediction. Generic bacterial, archaebacterial, or combined parameters can be used.
Minimal gene number	If the number of predicted genes is more than given by this parameter then automatic training of gene finding parameters is involved and genes are repredicted based on automatically generated parameters. Default value is 50, minimal value is 1.
Minimal gene length	Minimal length of predicted genes in nucleotides. Default value is 60, minimal value is 10.
Do not predict promoters/terminators	Do not predict promoters/terminators.
Do not add sequence name	Do not add sequence name to ID lines of predicted genes/proteins.

FgenesV

Trained Pattern/Markov chain-based viral gene prediction

FgenesV algorithm is based on pattern recognition of different types of signals and Markov chain models of coding regions. Optimal combination of these features is then found by dynamic programming and a set of gene models is constructed along given sequence.
FgenesV is the fastest *ab initio* viral gene prediction program available.

We developed new **FgenesV-Annotator** script that finds similar proteins in public databases and annotates predicted genes. This script can also identify low scoring genes if they have known homologous protein.

As an example of using FgenesV, the annotation of *SARS coronavirus TOR2 genome* is presented:
Annotation of complete genome of the SARS associated Coronavirus FgenesV-Annotator script.

There are two variants of viral gene prediction program: FgenesV0, which is suited for small (<10 kb) genomes, uses generic parameters of coding regions, while FgenesV learns genome-specific parameters using viral genome sequence as an input.

FgenesV predicts all intronless viral genes. To find small group of genes that contain introns - normally alternative structures of intronless variants - standard eukaryotic gene finding programs, such as **Fgenesh** , can be used in addition to FgenesV.

As additional parameters, you can choose Linear or Circular form of your virus and select alternative genetic code (Standard code is default): The Bacterial and Plant Plastid Code (transl_table=11) or The Mold, Protozoan, and Coelenterate Mitochondrial Code and the Mycoplasma/Spiroplasma Code (transl_table=4).

Parameters:

Input	
Sequences set	Input file.
Output	
Result	Name of the output file

FgenesV0

Generic parameters Markov chain-based viral gene prediction

FgenesV algorithm is based on pattern recognition of different types of signals and Markov chain models of coding regions. Optimal combination of these features is then found by dynamic programming and a set of gene models is constructed along given sequence.

FgenesV is the fastest *ab initio* viral gene prediction program available.

We developed new **FgenesV-Annotator** script that finds similar proteins in public databases and annotates predicted genes. This script can also identify low scoring genes if they have known homologous protein.

As an example of using FgenesV, the annotation of *SARS coronavirus TOR2 genome* is presented:

Annotation of complete genome of the SARS associated Coronavirus FgenesV-Annotator script.

There are two variants of viral gene prediction program: FgenesV0, which is suited for small (<10 kb) genomes, uses generic parameters of coding regions, while Fgenesv learns genome-specific parameters using viral genome sequence as an input.

FgenesV predicts all intronless viral genes. To find small group of genes that contain introns - normally alternative structures of intronless variants - standard eukaryotic gene finding programs, such as **Fgenesh** , can be used in addition to FgenesV.

As additional parameters, you can choose Linear or Circular form of your virus and select alternative genetic code (Standard code is default): The Bacterial and Plant Plastid Code (transl_table=11) or The Mold, Protozoan, and Coelenterate Mitochondrial Code and the Mycoplasma/Spiroplasma Code (transl_table=4).

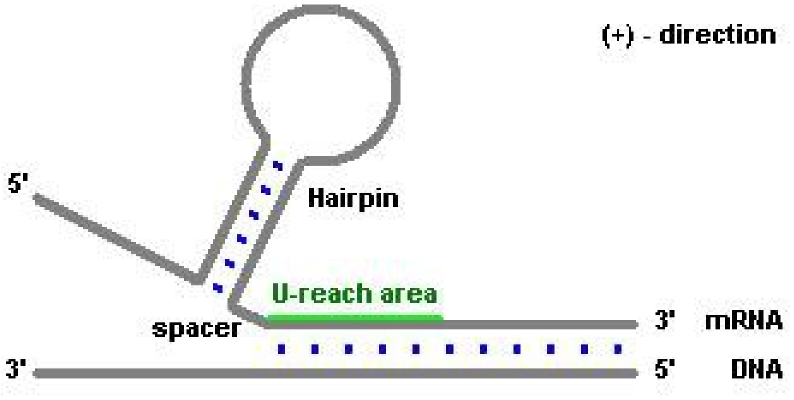
Parameters:

Input	
Sequence	Input file.
Output	
Result	Name of the output file

FindTerm

FindTerm - a program for searching bacterial terminators in DNA sequences. The set of conditions for searching bacterial terminators is stored in the config file.

Scheme of transcription



This scheme corresponds to positive direction (+) of tranccription form 3' to 5' end of DNA, and when we search terminators oriented from 5' to 3' end, found structure will be marked by (-) in the output file (see below).

First the program searches for region, which meets the requirements for T-reach region. Then it tries possible combinations of spacer lengths. At last, it finds all hairpins which meet user-defined parameters and complementarity rules. Then it searches the next appropriate T-reach region. Structures which meet all requirements are displayed.

```
Output and representing the results
There are examples of FindTerm output:

FindTerm - search for Rho-independent bacterial terminators
(Softberry, 2004)
Mode: All non-overlapping
Chain Start Length   Score
  -       2     33   -22.9
  +      93     53   -33.1
  -     210     52   -33.3
  +     315     53   -37.5
  +     423     53   -24.8

or

FindTerm - search for Rho-independent bacterial terminators
(Softberry, 2004)
Mode: Best terminator
Chain Start Length   Score
  +     423     53   -37.5

<Chain>  indicates the chain direction:
         (+) means that terminator is oriented from 3' to 5' end of DNA
         (-) means that terminator is oriented from 5' to 3' end of DNA
<Start>  is the position at which terminator begins
<Length> is the length of terminator, from the start of hairpin and up to end
of T-reach region
<Score>  is the value of score function, including enegy of terminator.
         The lower Score corresponds to the better terminator.
```

Parameters:

Input	
Sequence	Findterm Input file.
Output	
Result	Name of the output file.
XML data	Name of the file for graphical output.
Options	

Energy threshold value	Energy threshold value (default value is -11.0, minimal value is -100, maximal value is 100). Accounts for stem energy, sequence similarity with the known terminators etc.
Work modes	Defines one of 2 working modes: **Best terminator** - only best terminator at output **All non-overlapping terminators** - Output all non-overlapping terminators in both "+" and "-" chains at once, which are not closer than 20 nucleotides to each other.

Gene Finding

BestORF

Prediction of potential coding fragments in EST/mRNA sequence.

Method description:

Algorithm is based on Markov chain model of coding regions and a probabilistic model to combine it with Start codon potential.

Accuracy:

Our tests show that accuracy of frame recognition (true ORF) is about 100% for typical mRNA and about 99% for mRNA fragments of 500 - 800 bp containing partial coding region. Accuracy is lower for EST with frameshift errors, or for EST with very short coding fragments.

The program outputs potential CDS positions produced taking into account probabilities of each potential start codon, as well as longest ORF positions, as an extension of CDS upstream from start codon). If all observed Met codons are recognized as internal, i.e. if predicted translation start codon is missing from the sequence, CDS and ORF have the same positions.

```
Example of Output:
BestORF  Prediction of potential coding fragment in plant EST/mRNA sequence
 Time:   Tue Feb 16 20:03:57 1999.
 Seq name: Seq_name:
 Length of sequence:  388
 Predicted CDS 1 in +chain 1 in -chain 0
 Position of predicted CDS/ORF:
  G Str Feature   Start      End   Score       ORF       CDS-Len Frame

  1 +   1 CDSo      30 -     386   30.57      3 -    386    357     +3

Predicted protein fragment:
>BestORF   1   1 fragment (s)     30  -    386    119 aa, chain +
MDELDILIVGGYWGKGSRGGMMSHFLCAVAEKPPPGEKPSVFHTLSRVGSGCTMKELYDL
GLKLAKYWKPFHRKAPPSSILCGTEKPEVYIEPCNSVIVQIKAAEIVPSDMYKTGCTLR
```

Abbreviations: G - gene (CDS/ORF), Str - Strand, CDS-Len - CDS Length.

Parameters:

Input	
Organism	Parameter file for specified organizm
Sequences	File with nucleotide sequences in FASTA format
Output	
Result file	Name of the output file

Fex

Prediction of internal, 5'- and 3'- exons in Human DNA sequences.

Method description:

Algorithm first predicts all internal exons in a given sequence by linear discriminant function combining characteristics describing donor and acceptor splice sites, 5'- and 3'-intron regions and also coding regions for each open reading frame flanked by GT and AG base pairs. Potential 5'- and 3'- exons are predicted by corresponding discriminant functions on the left side of the first internal exon and on the right side from last internal exon, respectively.

Accuracy:

The accuracy of precise exon recognition on the set of 210 genes (with 761 internal exons) is 70% with a specificity of 63%. The recognition quality computed at the level of individual nucleotides is 87% for exons sequences (Sp=82%) with the level 97% for intron sequences. This program does not assemble the exons and is more reliable for a case of missing exons - for example, due to sequencing errors.

Fex output:

First line - name of your sequence

Next lines - positions of predicted exons, their 'weights', ORF number and potential number ORFs for a particular exon.

For example:

```
Seq name: Adh_and_cact.1 (2919020 bases) 848501 853000
 Length of sequence:  4500  Exon thr-  0 Overlap thr-    0.0
 # of potential exons: 9
    2758 -    2936 + w=   27.96 ORF= 0 First    exon     2758 -    2934
    3291 -    3354 - w=   13.63 ORF= 2 First    exon     3292 -    3354
    2577 -    2690 + w=   11.78 ORF= 2 Internal exon     2579 -    2689
       3 -     269 + w=   10.06 ORF= 0 Single   exon        3 -     269
    3024 -    3107 - w=    9.15 ORF= 2 Internal exon     3025 -    3105
     385 -     543 + w=    2.22 ORF= 0 Last     exon      385 -     543
    3169 -    3173 + w=    2.18 ORF= 0 First    exon     3169 -    3171
    2213 -    2380 + w=    1.65 ORF= 0 Last     exon     2213 -    2380
    1037 -    1076 + w=    0.25 ORF= 0 First    exon     1037 -    1075
>Exon-     1 Amino acid sequence -    59 aa, chain +
MANCPHTIGVEFGTRIIEVDDKKIKLQIWDTAGQERFRAVTRSYYRGAAGALMVYDITR
>Exon-     2 Amino acid sequence -    21 aa, chain -
MACAELRTRRRSDRADPPGCS
>Exon-     3 Amino acid sequence -    37 aa, chain +
PNMTAAPYNYNYIFKYIIIGDMGVGKSCLLHQFTEKK
>Exon-     4 Amino acid sequence -    88 aa, chain +
MLVQTPGISKSWMSSICLRESTFFMSCDRFRRSVSHCEGDTHELTAWQRVYLATHIWHRL
AGAQVVDLHIVNFVYEHLEGRFLLKIKT
>Exon-     5 Amino acid sequence -    27 aa, chain -
NLPSALQIRFVANEKDHSAGIGEIASV
>Exon-     6 Amino acid sequence -    52 aa, chain +
CDRRKPSKTRERKSSEKRLLICIDLPIENNRNNCLSVQPRNPAKPVCVLARK
>Exon-     7 Amino acid sequence -     1 aa, chain +
M
>Exon-     8 Amino acid sequence -    55 aa, chain +
LAGKQTRSAVQTQAGLKKKYRGQFEKGEQNVVSTQNKLMQRLGLLISSDYGWTFK
>Exon-     9 Amino acid sequence -    13 aa, chain +
MVGQKRPPLYLKI
```

References:

Solovyev V.V.,Salamov A.A., Lawrence C.B. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. (Nucl.Acids Res.,1994,22,24,5156-5163).

Solovyev V.V., Salamov A.A. , Lawrence C.B. The prediction of human exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. in: The Second International conference on Intelligent systems for Molecular Biology (eds. Altman R., Brutlag D., Karp R., Latrop R. and Searls D.), AAAI Press, Menlo Park, CA (1994, 354-362).

Parameters:

Input	
Organism	Select parameter file for specified organizm.
Input file	Browse your source file with nucleotide sequences in FASTA format.
Output	
Output file	Name of the output file.

Fgenes

Pattern based human gene structure prediction (multiple genes, both chains).

Method description:

Algorithm based on pattern recognition of different types of exons, promoters and polyA signals. Optimal combination of these features is then found by dynamic programming and a set of gene models is constructed along a given sequence.

Fgenes output:

G - predicted gene number, starting from start of sequence;
Str - DNA strand (+ for direct and - for complementary strands);
Feature - type of coding sequence: CDSf - First (Starting with Start codon), CDSi - internal (internal exon), CDSl - last coding segment, ending with stop codon);
TSS - position of transcription start;
TATA – position of TATA-box;
wTATA – Discriminant function score for TATA box;
TSS - Positions of transcription start (TATA-box position and score);
Start and End - Position of the Feature;
Weight - Discriminant function score for the feature;
ORF - start/end positions of ORF where the first complete codon starts and the last codon ends.

```
FGENES 1.5 Prediction of multiple genes in genomic DNA
Time: 171940.7 Date: 20001003
Seq name: > HUMHBB      73308 bp    DNA            PRI       20-JAN-1
Length of sequence:   73308 GC content: 0.39 Zone: 1
Number of predicted genes:   9 In +chain:   7 In -chain:   2
Number of predicted exons:  23 In +chain:  19 In -chain:   4
Positions of predicted genes and exons:
 G Str Feature  Start       End   Weight  ORF-start ORF-end

 1 -   1 CDSi    5978 -    6039    1.69    5978 -    6037
 1 -   2 CDSf    6314 -    6365    1.40    6315 -    6365

 2 -   1 CDSl   13709 -   13807    1.84   13712 -   13807
 2 -   2 CDSf   14781 -   14855    1.62   14781 -   14855

 3 +     TSS    19488              5.83 TATA  19457 wTATA   19.85 LDF   0.81
 3 +   1 CDSf   19541 -   19632   11.08   19541 -   19630
 3 +   2 CDSi   19755 -   19977    6.20   19756 -   19977
 3 +   3 CDSl   20833 -   20961    5.95   20833 -   20958
 3 +     PolA   21055              2.08

 4 +     TSS    34478              4.98 TATA  34447 wTATA   19.21 LDF   0.91
 4 +   1 CDSf   34531 -   34622    8.82   34531 -   34620
 4 +   2 CDSi   34745 -   34967    5.96   34746 -   34967
 4 +   3 CDSl   35854 -   35982    6.30   35854 -   35979
 4 +     PolA   36043              2.68

 5 +     TSS    39412              5.00 TATA  39383 wTATA   19.21 LDF   0.93
 5 +   1 CDSf   39467 -   39558    8.82   39467 -   39556
 5 +   2 CDSi   39681 -   39903    5.96   39682 -   39903
 5 +   3 CDSl   40770 -   40898    6.17   40770 -   40895
 5 +     PolA   40959              2.78

 6 +   1 CDSf   45995 -   46151    3.09   45995 -   46150
 6 +   2 CDSl   46997 -   47100    2.32   46999 -   47097
 6 +     PolA   47243              2.75

 7 +   1 CDSf   54790 -   54881    8.97   54790 -   54879
 7 +   2 CDSi   55010 -   55232    5.60   55011 -   55232
 7 +   3 CDSl   56131 -   56259    5.05   56131 -   56256
 7 +     PolA   56365              1.07

 8 +   1 CDSf   62187 -   62278    9.72   62187 -   62276
 8 +   2 CDSi   62409 -   62631    6.64   62410 -   62631
```

```
  8 +   3 CDSl   63482 -   63610    6.56   63482 -   63607
  8 +     PolA   63718              4.72

  9 +   1 CDSf   68183 -   68290    2.50   68183 -   68290
  9 +   2 CDSl   70703 -   70819    1.10   70703 -   70816
  9 +     PolA   70905              4.71

Predicted proteins:
>FGENES 1.5 > HUMHBB      7   1 Multiexon gene    5978 -    6365      38 a Ch-
MVCNCGLDHNFQSPRSKTCAFNKLIYTTSTLGSSSINE
>FGENES 1.5 > HUMHBB      7   2 Multiexon gene   13709 -   14855      57 a Ch-
MCSHHLASNCCFRSVPLPHLSRSLQEFVLKVNFHNRKLIEAKASVKERNISSKPLCC
>FGENES 1.5 > HUMHBB      7   3 Multiexon gene   19541 -   20961     147 a Ch+
MVHFTAEEKAAVTSLWSKMNVEEAGGEALGRLLVVYPWTQRFFDSFGNLSSPSAILGNPK
VKAHGKKVLTSFGDAIKNMDNLKPAFAKLSELHCDKLHVDPENFKLLGNVMVIILATHFG
KEFTPEVQAAWQKLVSAVAIALAHKYH
>FGENES 1.5 > HUMHBB      7   4 Multiexon gene   34531 -   35982     147 a Ch+
MGHFTEEDKATITSLWGKVNVEDAGGETLGRLLVVYPWTQRFFDSFGNLSSASAIMGNPK
VKAHGKKVLTSLGDAIKHLDDLKGTFAQLSELHCDKLHVDPENFKLLGNVLVTVLAIHFG
KEFTPEVQASWQKMVTGVASALSSRYH
>FGENES 1.5 > HUMHBB      7   5 Multiexon gene   39467 -   40898     147 a Ch+
MGHFTEEDKATITSLWGKVNVEDAGGETLGRLLVVYPWTQRFFDSFGNLSSASAIMGNPK
VKAHGKKVLTSLGDAIKHLDDLKGTFAQLSELHCDKLHVDPENFKLLGNVLVTVLAIHFG
KEFTPEVQASWQKMVTAVASALSSRYH
>FGENES 1.5 > HUMHBB      7   6 Multiexon gene   45995 -   47100      86 a Ch+
MGNPKVKAHGKKVLISFGKAVMLTDDLKGTFATLSDLHCNKLHVDPENFLVSTLRQRDID
CFGNPLQRGFYPTDTGFLAVTNKCCG
>FGENES 1.5 > HUMHBB      7   7 Multiexon gene   54790 -   56259     147 a Ch+
MVHLTPEEKTAVNALWGKVNVDAVGGEALGRLLVVYPWTQRFFESFGDLSSPDAVMGNPK
VKAHGKKVLGAFSDGLAHLDNLKGTFSQLSELHCDKLHVDPENFRLLGNVLVCVLARNFG
KEFTPQMQAAYQKVVAGVANALAHKYH
>FGENES 1.5 > HUMHBB      7   8 Multiexon gene   62187 -   63610     147 a Ch+
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK
VKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG
KEFTPPVQAAYQKVVAGVANALAHKYH
>FGENES 1.5 > HUMHBB      7   9 Multiexon gene   68183 -   70819      74 a Ch+
MEQSWAENDFDELREEGFRRSNYSKLKEEVRTNGKEASIILIPKPDRDTTKKENVTPISL
MNIDAKILNKILAN
```

Parameters:

Input	
Sequences	File with nucleotide sequences in FASTA format
Output	
Result file	Name of the output file

Fgenes-m

Pattern-based prediction of multiple variants of gene structure.

There are two reasons to predict several sub-optimal variants of gene structure, instead of only one:

1) Gene prediction algorithms for long genomic sequences are only 70-80% accurate on average, therefore real gene structure might have the score slightly lower than the predicted optimal variant. Fgenes-m allows you to see alternative structures that otherwise you might never see; and

2) Alternative splicing is quite common for mammalian genes, so you may miss real gene structures relying on just one optimal prediction, even supported by experimental data.

Of course, thousands of alternative gene structures can be predicted, and there is currently no established way to distinguish true variants from false ones.

Fgenes-m variant proved to be useful in providing a set of possible gene structures for further experimental testing in commercial gene hunting.

Method description:

Algorithm outputs several (up to 15, though the number can be changed) suboptimal variants of predicted gene structure. It is similar to Fgenes and is based on pattern recognition of different types of exons, promoters and polyA signals and finding optimal combination of them by dynamic programming. Then, a set of gene models along given sequences is constructed.

You may compare validities of predicted variants using GENE WEIGHT parameter. If this parameter is similar in alternative variants, it is reasonable to consider them.

Fgenes-M output:

```
 FGENES-M 1.5.0 Prediction of several variants of  multiple genes
 Time: 175701.1 Date: 19981005
 Seq name:  ACU08131
 Length of sequence:    5392 GC content: 0.46 Zone: 2
 Number of predicted genes:   1 In +chain:   1 In -chain:   0
 Number of predicted exons:   6 In +chain:   6 In -chain:   0
 Predicted genes and exons in var:   1 Max var=   10 GENE WEIGHT:   24.1
  G Str Feature  Start       End   Weight  ORF-start ORF-end

  1 +     TSS      355              7.43 TATA    327 wTATA   21.08 LDF   0.56
  1 +   1 CDSf      521 -     641    1.23     521 -     640
  1 +   2 CDSi     1066 -    1362    2.08    1068 -    1361
  1 +   3 CDSi     1860 -    2028    1.69    1862 -    2026
  1 +   4 CDSi     2637 -    2802    2.74    2638 -    2802
  1 +   5 CDSi     3558 -    3797    4.35    3558 -    3797
  1 +   6 CDSl     4131 -    4247    2.09    4131 -    4244
  1 +     PolA    4650              3.17

Predicted proteins:
>FGENES-M 1.5  ACU08131         1 Multiexon gene     521 -    4247     369 a
Ch+
MAGTVTEAWDVAVFAARRRNDEDDTTRDSLFTYTNSNNTRGPFEGPNYHIAPRWVYNITS
VWMIFVVIASIFTNGLVLVATAKFKKLRHPLNWILVNLAIADLGETVIASTISVINQISG
YFILGHPMCVLEGYTVSTCGISALWSLAVISWERWVVVCKPFGNVKFDAKLAVAGIVFSW
VWSAVWTAPPVFGWSRYWPHGLKTSCGPDVFSGSDDPGVLSYMIVLMITCCFIPLAVILL
CYLQVWLAIRAVAAQQKESESTQKAEKEVSRMVVVMIIAYCFCWGPYTVFACFAAANPGY
AFHPLAAALPAYFAKSATIYNPIIYVFMNRQFRNCIMQLFGKKVDDGSELSSTSRTEVSS
VSNSSVSPA
 FGENES-M 1.5.0 Prediction of several variants of  multiple genes
 Time: 175701.1 Date: 19981005
 Seq name:  ACU08131
 Length of sequence:    5392 GC content: 0.46 Zone: 2
 Number of predicted genes:   1 In +chain:   1 In -chain:   0
 Number of predicted exons:   6 In +chain:   6 In -chain:   0
 Predicted genes and exons in var:   2 Max var=   10 GENE WEIGHT:   15.1
  G Str Feature  Start       End   Weight  ORF-start ORF-end

  1 +   1 CDSf      218 -     321    1.01     218 -     319
  1 +   2 CDSi      984 -    1023    1.94     986 -    1021
  1 +   3 CDSi     1860 -    2028    1.49    1862 -    2026
  1 +   4 CDSi     2675 -    2802    1.00    2676 -    2801
  1 +   5 CDSi     3558 -    3797    4.35    3558 -    3797
  1 +   6 CDSl     4131 -    4247    2.09    4131 -    4244
  1 +     PolA    4650              3.17

Predicted proteins:
>FGENES-M 1.5  ACU08131         1 Multiexon gene     218 -    4247     265 a
Ch+
MRQGGGQITAQLRDKTFKGFEDLVLQVRGLIRLGGNLLVDVCVVIAILVSQLSGPWPLYL
GNAGSLSASPLEMSSSMPNWPWLALSSPGCGLLYGQHHPSLAGVDVFSGSDDPGVLSYMI
VLMITCCFIPLAVILLCYLQVWLAIRAVAAQQKESESTQKAEKEVSRMVVVMIIAYCFCW
GPYTVFACFAAANPGYAFHPLAAALPAYFAKSATIYNPIIYVFMNRQFRNCIMQLFGKKV
DDGSELSSTSRTEVSSVSNSSVSPA
 FGENES-M 1.5.0 Prediction of several variants of  multiple genes
 Time: 175701.1 Date: 19981005
 Seq name:  ACU08131
 Length of sequence:    5392 GC content: 0.46 Zone: 2
 Number of predicted genes:   1 In +chain:   1 In -chain:   0
 Number of predicted exons:   6 In +chain:   6 In -chain:   0
 Predicted genes and exons in var:   3 Max var=   10 GENE WEIGHT:   14.3
  G Str Feature  Start       End   Weight  ORF-start ORF-end

  1 +     TSS      355              7.43 TATA    327 wTATA   21.08 LDF   0.56
  1 +   1 CDSf     521 -     641    1.23     521 -     640
  1 +   2 CDSi    1066 -    1362    2.08    1068 -    1361
  1 +   3 CDSi    1860 -    2028    1.69    1862 -    2026
  1 +   4 CDSi    2637 -    2802    2.74    2638 -    2802
  1 +   5 CDSi    3558 -    3870    0.78    3558 -    3869
  1 +   6 CDSl    4857 -    5131    2.37    4859 -    5128
  1 +     PolA    5187              0.77

Predicted proteins:
>FGENES-M 1.5  ACU08131         1 Multiexon gene     521 -    5131     446 a
Ch+
MAGTVTEAWDVAVFAARRRNDEDDTTRDSLFTYTNSNNTRGPFEGPNYHIAPRWVYNITS
VWMIFVVIASIFTNGLVLVATAKFKKLRHPLNWILVNLAIADLGETVIASTISVINQISG
YFILGHPMCVLEGYTVSTCGISALWSLAVISWERWVVVCKPFGNVKFDAKLAVAGIVFSW
VWSAVWTAPPVFGWSRYWPHGLKTSCGPDVFSGSDDPGVLSYMIVLMITCCFIPLAVILL
CYLQVWLAIRAVAAQQKESESTQKAEKEVSRMVVVMIIAYCFCWGPYTVFACFAAANPGY
AFHPLAAALPAYFAKSATIYNPIIYVFMNRQVIFCVPKWTVTGLARRVQKREGCMVFTGA
RECIEGGQEEEKFVPRGVCASAKSNALNLNSVESGHDSDTGRTNETQHDPPRSLQGLCAS
SQHGSTGTILYIVFDTKACCVPGTSS
 FGENES-M 1.5.0 Prediction of several variants of  multiple genes
 Time: 175701.1 Date: 19981005
 Seq name:  ACU08131
 Length of sequence:    5392 GC content: 0.46 Zone: 2
 Number of predicted genes:   1 In +chain:   1 In -chain:   0
 Number of predicted exons:   6 In +chain:   6 In -chain:   0
 Predicted genes and exons in var:   4 Max var=   10 GENE WEIGHT:   13.9
  G Str Feature  Start       End   Weight  ORF-start ORF-end

  1 +     TSS      355              7.43 TATA    327 wTATA   21.08 LDF   0.56
  1 +   1 CDSf     521 -     641    1.23     521 -     640
  1 +   2 CDSi    1066 -    1362    2.08    1068 -    1361
  1 +   3 CDSi    1860 -    2028    1.69    1862 -    2026
  1 +   4 CDSi    2637 -    2802    2.74    2638 -    2802
  1 +   5 CDSi    3558 -    3668    0.99    3558 -    3668
  1 +   6 CDSl    4131 -    4247    2.09    4131 -    4244
  1 +     PolA    4650              3.17

Predicted proteins:
>FGENES-M 1.5  ACU08131         1 Multiexon gene     521 -    4247     326 a
Ch+
MAGTVTEAWDVAVFAARRRNDEDDTTRDSLFTYTNSNNTRGPFEGPNYHIAPRWVYNITS
VWMIFVVIASIFTNGLVLVATAKFKKLRHPLNWILVNLAIADLGETVIASTISVINQISG
YFILGHPMCVLEGYTVSTCGISALWSLAVISWERWVVVCKPFGNVKFDAKLAVAGIVFSW
VWSAVWTAPPVFGWSRYWPHGLKTSCGPDVFSGSDDPGVLSYMIVLMITCCFIPLAVILL
CYLQVWLAIRAVAAQQKESESTQKAEKEVSRMVVVMIIAYCFCWGPYTFRNCIMQLFGKK
VDDGSELSSTSRTEVSSVSNSSVSPA
 FGENES-M 1.5.0 Prediction of several variants of  multiple genes
 Time: 175701.1 Date: 19981005
 Seq name:  ACU08131
 Length of sequence:    5392 GC content: 0.46 Zone: 2
 Number of predicted genes:   1 In +chain:   1 In -chain:   0
 Number of predicted exons:   5 In +chain:   5 In -chain:   0
 Predicted genes and exons in var:   5 Max var=   10 GENE WEIGHT:   13.0
  G Str Feature  Start       End   Weight  ORF-start ORF-end

  1 +     TSS      355              7.43 TATA    327 wTATA   21.08 LDF   0.56
  1 +   1 CDSf     521 -     641    1.23     521 -     640
  1 +   2 CDSi    1066 -    1362    2.08    1068 -    1361
  1 +   3 CDSi    1860 -    2028    1.69    1862 -    2026
  1 +   4 CDSi    2637 -    2802    2.74    2638 -    2802
  1 +   5 CDSl    3558 -    3875    2.10    3558 -    3872
  1 +     PolA    4650                3.17

Predicted proteins:
>FGENES-M 1.5  ACU08131         1 Multiexon gene     521 -    3875     356 a
Ch+
MAGTVTEAWDVAVFAARRRNDEDDTTRDSLFTYTNSNNTRGPFEGPNYHIAPRWVYNITS
VWMIFVVIASIFTNGLVLVATAKFKKLRHPLNWILVNLAIADLGETVIASTISVINQISG
YFILGHPMCVLEGYTVSTCGISALWSLAVISWERWVVVCKPFGNVKFDAKLAVAGIVFSW
VWSAVWTAPPVFGWSRYWPHGLKTSCGPDVFSGSDDPGVLSYMIVLMITCCFIPLAVILL
CYLQVWLAIRAVAAQQKESESTQKAEKEVSRMVVVMIIAYCFCWGPYTVFACFAAANPGY
AFHPLAAALPAYFAKSATIYNPIIYVFMNRQVIFCVPKWTVTGLARRVQKREGCMG
```

Parameters:

Input	
Sequence	Source file with nucleotide sequences in FASTA format.
Output	
Result file	Name of the output file.
Options	
Alternative genes	Count of alternative gene.

Fgenesh

Program for predicting multiple genes in genomic DNA sequences.

Fgenesh is the fastest (50-100 times faster than GenScan) and most accurate gene finder available (see: Figure and Table, respectively). In recent rice genome sequencing projects, it was cited "the most successful (gene finding) program (Yu *et al*. (2002) Science 296:79) and was used to produce 87% of all high-evidence predicted genes (Goff *et al*. (2002) Science 296:79).

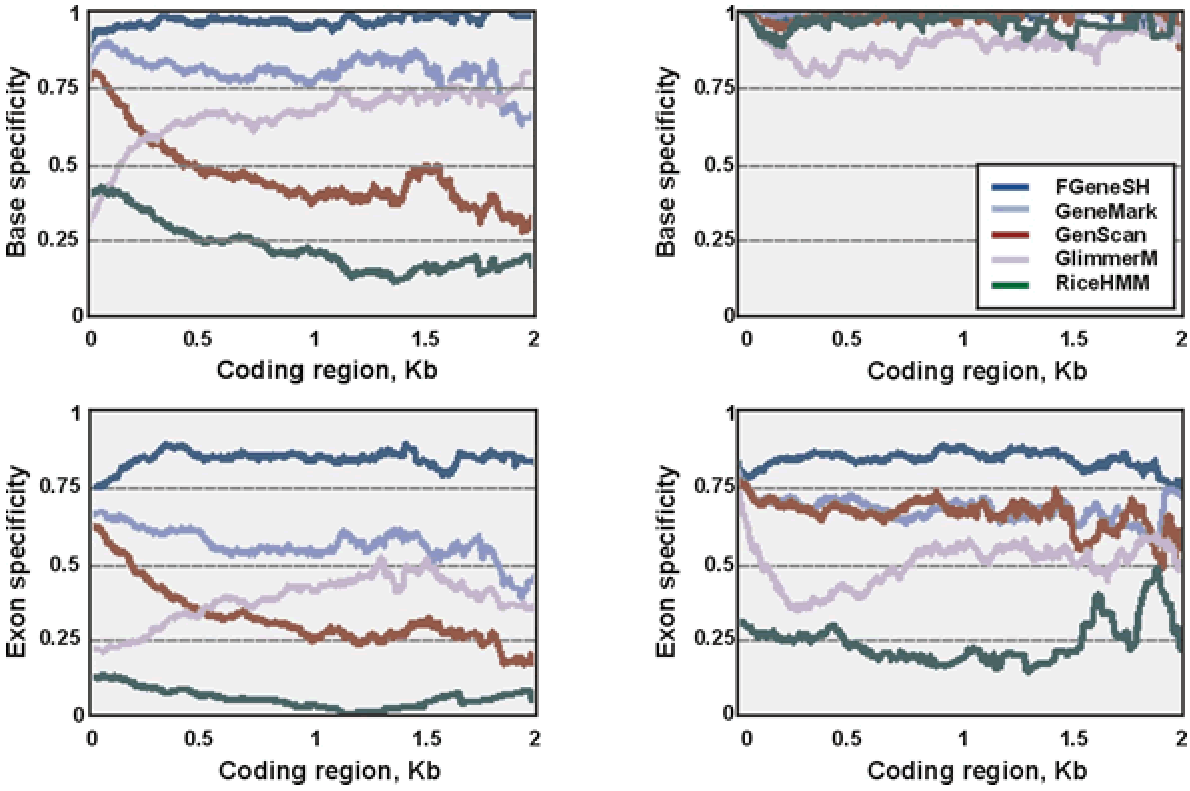


Figure. Performance of different gene finding programs on rice genes (reprinted from Yu et al., 2002, Science, 296:79-92). These tests confirmed that Fgenesh is by far the most accurate program (of five programs tested).

Table. Performance of three popular gene prediction programs on 42 semi-artificial genomic sequences containing 178 known human gene sequences (900 exons). Sensitivity is percentage of exons that are predicted correctly. Selectivity is percentage of predicted exons that are correct (these results reproduced with some changes from Yada et al., 2002, Cold Spring Harbor Genome Sequencing and Biology Meeting, May 7-11). These tests demonstrated that Fgenesh is by far the most accurate program (of three programs tested).

Program	**Sensitivity**	**Specificity**	**Missed Exons, %**	**Wrong Exons, %**
Fgenesh	77.1	65.7	9.6	23.2
GenScan	66.5	44.9	12.0	40.9
HMMGene	69.6	36.6	15.5	55.5

Web version of Fgenesh can be used with parameters for the following genomes: human, mouse, Drosophila, nematode, dicot plants, monocot plants, yeast (S.pombe) and Neurospora.

Check appropriate genome/organism and Fgenesh program. Paste your sequence to the window or load your file with sequence in FASTA format and click *Perform Search* button.

References:

Salamov A., Solovyev V. (2000) Ab initio gene finding in Drosophila genomic DNA. Genome Res., 10,516-522

Fgenesh output:

```
FGENESH 2.6 Prediction of potential genes in Homo_sapiens genomic DNA
 Time    :   Thu Dec 27 19:47:24 2007
  Seq  name:  gi|13907843|ref|NG_000007.1|  Homo  sapiens  genomic  beta  globin
region (HBB@) on chromosome 11
 Length of sequence: 73308
 Number of predicted genes 10: in +chain 10, in -chain 0.
 Number of predicted exons 21: in +chain 21, in -chain 0.
```

```
  Positions   of   predicted   genes   and   exons:   Variant      1   from      1,
Score:180.171899
   G Str   Feature   Start       End    Score          ORF           Len

   1 +      TSS      19456               -7.09
   1 +    1 CDSf     19541 -     19632   16.13     19541 -     19630     90
   1 +    2 CDSi     19755 -     19977   13.37     19756 -     19977    222
   1 +    3 CDSl     20833 -     20961    3.34     20833 -     20961    129
   1 +      PolA     21055                1.13

   2 +      TSS      34446               -7.09
   2 +    1 CDSf     34531 -     34622   13.42     34531 -     34620     90
   2 +    2 CDSi     34745 -     34967   21.52     34746 -     34967    222
   2 +    3 CDSl     35854 -     35982    2.92     35854 -     35982    129
   2 +      PolA     36043                1.13

   3 +      TSS      39382               -7.09
   3 +    1 CDSf     39467 -     39558   13.42     39467 -     39556     90
   3 +    2 CDSi     39681 -     39903   21.52     39682 -     39903    222
   3 +    3 CDSl     40770 -     40898    3.66     40770 -     40898    129
   3 +      PolA     40959                1.13

   4 +      TSS      44415               -8.69
   4 +    1 CDSf     45995 -     46151   16.58     45995 -     46150    156
   4 +    2 CDSl     46997 -     47100   -1.94     46999 -     47100    102
   4 +      PolA     47243                1.13

   5 +      TSS      54707               -4.39
   5 +    1 CDSf     54790 -     54881   13.44     54790 -     54879     90
   5 +    2 CDSi     55010 -     55232   17.01     55011 -     55232    222
   5 +    3 CDSl     56425 -     56535    2.53     56425 -     56535    111
   5 +      PolA     56931                1.13

   6 +      TSS      62104               -6.59
   6 +    1 CDSf     62187 -     62278   12.99     62187 -     62276     90
   6 +    2 CDSi     62409 -     62631   20.06     62410 -     62631    222
   6 +    3 CDSl     63482 -     63610    9.54     63482 -     63610    129
   6 +      PolA     63718                1.13

   7 +      TSS      68088               -9.39
   7 +    1 CDSo     68183 -     68428   19.52     68183 -     68428    246
   7 +      PolA     68509                1.13

   8 +      TSS      69336              -10.29
   8 +    1 CDSo     69467 -     70072   16.45     69467 -     70072    606
   8 +      PolA     70131               -1.08

   9 +      TSS      70224              -12.49
   9 +    1 CDSo     70355 -     70819   17.10     70355 -     70819    465
   9 +      PolA     70905                1.13

  10 +      TSS      72085               -6.39
  10 +    1 CDSo     72135 -     72395    7.31     72135 -     72395    261
  10 +      PolA     72952                1.13

Predicted protein(s):
>FGENESH:[mRNA]   1   3 exon (s)  19541  -  20961   444 bp, chain +
ATGGTGCATTTTACTGCTGAGGAGAAGGCTGCCGTCACTAGCCTGTGGAGCAAGATGAAT
GTGGAAGAGGCTGGAGGTGAAGCCTTGGGCAGACTCCTCGTTGTTTACCCCTGGACCCAG
AGATTTTTTGACAGCTTTGGAAACCTGTCGTCTCCCTCTGCCATCCTGGGCAACCCCAAG
GTCAAGGCCCATGGCAAGAAGGTGCTGACTTCCTTTGGAGATGCTATTAAAAACATGGAC
AACCTCAAGCCCGCCTTTGCTAAGCTGAGTGAGCTGCACTGTGACAAGCTGCATGTGGAT
CCTGAGAACTTCAAGCTCCTGGGTAACGTGATGGTGATTATTCTGGCTACTCACTTTGGC
AAGGAGTTCACCCCTGAAGTGCAGGCTGCCTGGCAGAAGCTGGTGTCTGCTGTCGCCATT
```

```
GCCCTGGCCCATAAGTACCACTGA
>FGENESH:[exon] Gene:  1 Exon:  1  Pos: 19541  -  19632    92 bp., chain +
ATGGTGCATTTTACTGCTGAGGAGAAGGCTGCCGTCACTAGCCTGTGGAGCAAGATGAAT
GTGGAAGAGGCTGGAGGTGAAGCCTTGGGCAG
>FGENESH:[exon] Gene:  1 Exon:  2  Pos: 19755  -  19977   223 bp., chain +
ACTCCTCGTTGTTTACCCCTGGACCCAGAGATTTTTTGACAGCTTTGGAAACCTGTCGTC
TCCCTCTGCCATCCTGGGCAACCCCAAGGTCAAGGCCCATGGCAAGAAGGTGCTGACTTC
CTTTGGAGATGCTATTAAAAACATGGACAACCTCAAGCCCGCCTTTGCTAAGCTGAGTGA
GCTGCACTGTGACAAGCTGCATGTGGATCCTGAGAACTTCAAG
>FGENESH:[exon] Gene:  1 Exon:  3  Pos: 20833  -  20961   129 bp., chain +
CTCCTGGGTAACGTGATGGTGATTATTCTGGCTACTCACTTTGGCAAGGAGTTCACCCCT
GAAGTGCAGGCTGCCTGGCAGAAGCTGGTGTCTGCTGTCGCCATTGCCCTGGCCCATAAG
TACCACTGA
>FGENESH:   1   3 exon (s)  19541  -  20961   147 aa, chain +
MVHFTAEEKAAVTSLWSKMNVEEAGGEALGRLLVVYPWTQRFFDSFGNLSSPSAILGNPK
VKAHGKKVLTSFGDAIKNMDNLKPAFAKLSELHCDKLHVDPENFKLLGNVMVIILATHFG
KEFTPEVQAAWQKLVSAVAIALAHKYH
>FGENESH:[mRNA]   2   3 exon (s)  34531  -  35982   444 bp, chain +
ATGGGTCATTTCACAGAGGAGGACAAGGCTACTATCACAAGCCTGTGGGGCAAGGTGAAT
GTGGAAGATGCTGGAGGAGAAACCCTGGGAAGGCTCCTGGTTGTCTACCCATGGACCCAG
AGGTTCTTTGACAGCTTTGGCAACCTGTCCTCTGCCTCTGCCATCATGGGCAACCCCAAA
GTCAAGGCACATGGCAAGAAGGTGCTGACTTCCTTGGGAGATGCCATAAAGCACCTGGAT
GATCTCAAGGGCACCTTTGCCCAGCTGAGTGAACTGCACTGTGACAAGCTGCATGTGGAT
CCTGAGAACTTCAAGCTCCTGGGAAATGTGCTGGTGACCGTTTTGGCAATCCATTTCGGC
AAAGAATTCACCCCTGAGGTGCAGGCTTCCTGGCAGAAGATGGTGACTGGAGTGGCCAGT
GCCCTGTCCTCCAGATACCACTGA
>FGENESH:[exon] Gene:  2 Exon:  1  Pos: 34531  -  34622    92 bp., chain +
ATGGGTCATTTCACAGAGGAGGACAAGGCTACTATCACAAGCCTGTGGGGCAAGGTGAAT
GTGGAAGATGCTGGAGGAGAAACCCTGGGAAG
>FGENESH:[exon] Gene:  2 Exon:  2  Pos: 34745  -  34967   223 bp., chain +
GCTCCTGGTTGTCTACCCATGGACCCAGAGGTTCTTTGACAGCTTTGGCAACCTGTCCTC
TGCCTCTGCCATCATGGGCAACCCCAAAGTCAAGGCACATGGCAAGAAGGTGCTGACTTC
CTTGGGAGATGCCATAAAGCACCTGGATGATCTCAAGGGCACCTTTGCCCAGCTGAGTGA
ACTGCACTGTGACAAGCTGCATGTGGATCCTGAGAACTTCAAG
>FGENESH:[exon] Gene:  2 Exon:  3  Pos: 35854  -  35982   129 bp., chain +
CTCCTGGGAAATGTGCTGGTGACCGTTTTGGCAATCCATTTCGGCAAAGAATTCACCCCT
GAGGTGCAGGCTTCCTGGCAGAAGATGGTGACTGGAGTGGCCAGTGCCCTGTCCTCCAGA
TACCACTGA
>FGENESH:   2   3 exon (s)  34531  -  35982   147 aa, chain +
MGHFTEEDKATITSLWGKVNVEDAGGETLGRLLVVYPWTQRFFDSFGNLSSASAIMGNPK
VKAHGKKVLTSLGDAIKHLDDLKGTFAQLSELHCDKLHVDPENFKLLGNVLVTVLAIHFG
KEFTPEVQASWQKMVTGVASALSSRYH
>FGENESH:[mRNA]   3   3 exon (s)  39467  -  40898   444 bp, chain +
ATGGGTCATTTCACAGAGGAGGACAAGGCTACTATCACAAGCCTGTGGGGCAAGGTGAAT
GTGGAAGATGCTGGAGGAGAAACCCTGGGAAGGCTCCTGGTTGTCTACCCATGGACCCAG
AGGTTCTTTGACAGCTTTGGCAACCTGTCCTCTGCCTCTGCCATCATGGGCAACCCCAAA
GTCAAGGCACATGGCAAGAAGGTGCTGACTTCCTTGGGAGATGCCATAAAGCACCTGGAT
GATCTCAAGGGCACCTTTGCCCAGCTGAGTGAACTGCACTGTGACAAGCTGCATGTGGAT
CCTGAGAACTTCAAGCTCCTGGGAAATGTGCTGGTGACCGTTTTGGCAATCCATTTCGGC
AAAGAATTCACCCCTGAGGTGCAGGCTTCCTGGCAGAAGATGGTGACTGCAGTGGCCAGT
GCCCTGTCCTCCAGATACCACTGA
>FGENESH:[exon] Gene:  3 Exon:  1  Pos: 39467  -  39558    92 bp., chain +
ATGGGTCATTTCACAGAGGAGGACAAGGCTACTATCACAAGCCTGTGGGGCAAGGTGAAT
GTGGAAGATGCTGGAGGAGAAACCCTGGGAAG
>FGENESH:[exon] Gene:  3 Exon:  2  Pos: 39681  -  39903   223 bp., chain +
GCTCCTGGTTGTCTACCCATGGACCCAGAGGTTCTTTGACAGCTTTGGCAACCTGTCCTC
TGCCTCTGCCATCATGGGCAACCCCAAAGTCAAGGCACATGGCAAGAAGGTGCTGACTTC
CTTGGGAGATGCCATAAAGCACCTGGATGATCTCAAGGGCACCTTTGCCCAGCTGAGTGA
ACTGCACTGTGACAAGCTGCATGTGGATCCTGAGAACTTCAAG
>FGENESH:[exon] Gene:  3 Exon:  3  Pos: 40770  -  40898   129 bp., chain +
CTCCTGGGAAATGTGCTGGTGACCGTTTTGGCAATCCATTTCGGCAAAGAATTCACCCCT
GAGGTGCAGGCTTCCTGGCAGAAGATGGTGACTGCAGTGGCCAGTGCCCTGTCCTCCAGA
TACCACTGA
>FGENESH:   3   3 exon (s)  39467  -  40898   147 aa, chain +
MGHFTEEDKATITSLWGKVNVEDAGGETLGRLLVVYPWTQRFFDSFGNLSSASAIMGNPK
VKAHGKKVLTSLGDAIKHLDDLKGTFAQLSELHCDKLHVDPENFKLLGNVLVTVLAIHFG
KEFTPEVQASWQKMVTAVASALSSRYH
>FGENESH:[mRNA]   4   2 exon (s)  45995  -  47100   261 bp, chain +
ATGGGCAACCCCAAAGTCAAGGCACATGGCAAGAAGGTGCTGATCTCCTTCGGAAAAGCT
GTTATGCTCACGGATGACCTCAAAGGCACCTTTGCTACACTGAGTGACCTGCACTGTAAC
AAGCTGCACGTGGACCCTGAGAACTTCCTGGTGAGTACTCTTAGGCAACGTGATATTGAT
TGTTTTGGCAACCCACTTCAGCGAGGATTTTACCCTACAGATACAGGCTTCTTGGCAGTA
ACTAACAAATGCTGTGGTTAA
>FGENESH:[exon] Gene:  4 Exon:  1  Pos: 45995  -  46151   157 bp., chain +
ATGGGCAACCCCAAAGTCAAGGCACATGGCAAGAAGGTGCTGATCTCCTTCGGAAAAGCT
GTTATGCTCACGGATGACCTCAAAGGCACCTTTGCTACACTGAGTGACCTGCACTGTAAC
AAGCTGCACGTGGACCCTGAGAACTTCCTGGTGAGTA
>FGENESH:[exon] Gene:  4 Exon:  2  Pos: 46997  -  47100   104 bp., chain +
CTCTTAGGCAACGTGATATTGATTGTTTTGGCAACCCACTTCAGCGAGGATTTTACCCTA
CAGATACAGGCTTCTTGGCAGTAACTAACAAATGCTGTGGTTAA
>FGENESH:   4   2 exon (s)  45995  -  47100    86 aa, chain +
MGNPKVKAHGKKVLISFGKAVMLTDDLKGTFATLSDLHCNKLHVDPENFLVSTLRQRDID
CFGNPLQRGFYPTDTGFLAVTNKCCG
>FGENESH:[mRNA]   5   3 exon (s)  54790  -  56535   426 bp, chain +
ATGGTGCATCTGACTCCTGAGGAGAAGACTGCTGTCAATGCCCTGTGGGGCAAAGTGAAC
GTGGATGCAGTTGGTGGTGAGGCCCTGGGCAGATTACTGGTGGTCTACCCTTGGACCCAG
AGGTTCTTTGAGTCCTTTGGGGATCTGTCCTCTCCTGATGCTGTTATGGGCAACCCTAAG
GTGAAGGCTCATGGCAAGAAGGTGCTAGGTGCCTTTAGTGATGGCCTGGCTCACCTGGAC
AACCTCAAGGGCACTTTTTCTCAGCTGAGTGAGCTGCACTGTGACAAGCTGCACGTGGAT
CCTGAGAACTTCAGGGTGTGTAAGAAGGTTCCTGAGGCTCTACAGATAGGGAGCACTTGT
TTATTTTACAAAGAGTACATGGGAAAAGAGAAAAGCAAGGGAACCGTACAAGGCATTAAT
GGGTGA
>FGENESH:[exon] Gene:  5 Exon:  1  Pos: 54790  -  54881    92 bp., chain +
ATGGTGCATCTGACTCCTGAGGAGAAGACTGCTGTCAATGCCCTGTGGGGCAAAGTGAAC
GTGGATGCAGTTGGTGGTGAGGCCCTGGGCAG
>FGENESH:[exon] Gene:  5 Exon:  2  Pos: 55010  -  55232   223 bp., chain +
ATTACTGGTGGTCTACCCTTGGACCCAGAGGTTCTTTGAGTCCTTTGGGGATCTGTCCTC
TCCTGATGCTGTTATGGGCAACCCTAAGGTGAAGGCTCATGGCAAGAAGGTGCTAGGTGC
CTTTAGTGATGGCCTGGCTCACCTGGACAACCTCAAGGGCACTTTTTCTCAGCTGAGTGA
GCTGCACTGTGACAAGCTGCACGTGGATCCTGAGAACTTCAGG
>FGENESH:[exon] Gene:  5 Exon:  3  Pos: 56425  -  56535   111 bp., chain +
GTGTGTAAGAAGGTTCCTGAGGCTCTACAGATAGGGAGCACTTGTTTATTTTACAAAGAG
TACATGGGAAAAGAGAAAAGCAAGGGAACCGTACAAGGCATTAATGGGTGA
>FGENESH:   5   3 exon (s)  54790  -  56535   141 aa, chain +
MVHLTPEEKTAVNALWGKVNVDAVGGEALGRLLVVYPWTQRFFESFGDLSSPDAVMGNPK
VKAHGKKVLGAFSDGLAHLDNLKGTFSQLSELHCDKLHVDPENFRVCKKVPEALQIGSTC
LFYKEYMGKEKSKGTVQGING
>FGENESH:[mRNA]   6   3 exon (s)  62187  -  63610   444 bp, chain +
ATGGTGCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAAGGTGAAC
GTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGGCTGCTGGTGGTCTACCCTTGGACCCAG
AGGTTCTTTGAGTCCTTTGGGGATCTGTCCACTCCTGATGCTGTTATGGGCAACCCTAAG
GTGAAGGCTCATGGCAAGAAAGTGCTCGGTGCCTTTAGTGATGGCCTGGCTCACCTGGAC
AACCTCAAGGGCACCTTTGCCACACTGAGTGAGCTGCACTGTGACAAGCTGCACGTGGAT
CCTGAGAACTTCAGGCTCCTGGGCAACGTGCTGGTCTGTGTGCTGGCCCATCACTTTGGC
AAAGAATTCACCCCACCAGTGCAGGCTGCCTATCAGAAAGTGGTGGCTGGTGTGGCTAAT
GCCCTGGCCCACAAGTATCACTAA
>FGENESH:[exon] Gene:  6 Exon:  1  Pos: 62187  -  62278    92 bp., chain +
ATGGTGCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAAGGTGAAC
GTGGATGAAGTTGGTGGTGAGGCCCTGGGCAG
>FGENESH:[exon] Gene:  6 Exon:  2  Pos: 62409  -  62631   223 bp., chain +
GCTGCTGGTGGTCTACCCTTGGACCCAGAGGTTCTTTGAGTCCTTTGGGGATCTGTCCAC
TCCTGATGCTGTTATGGGCAACCCTAAGGTGAAGGCTCATGGCAAGAAAGTGCTCGGTGC
CTTTAGTGATGGCCTGGCTCACCTGGACAACCTCAAGGGCACCTTTGCCACACTGAGTGA
GCTGCACTGTGACAAGCTGCACGTGGATCCTGAGAACTTCAGG
>FGENESH:[exon] Gene:  6 Exon:  3  Pos: 63482  -  63610   129 bp., chain +
CTCCTGGGCAACGTGCTGGTCTGTGTGCTGGCCCATCACTTTGGCAAAGAATTCACCCCA
CCAGTGCAGGCTGCCTATCAGAAAGTGGTGGCTGGTGTGGCTAATGCCCTGGCCCACAAG
TATCACTAA
```

```
>FGENESH:   6   3 exon (s)  62187  -  63610   147 aa, chain +
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK
VKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG
KEFTPPVQAAYQKVVAGVANALAHKYH
>FGENESH:[mRNA]   7   1 exon (s)  68183  -  68428   246 bp, chain +
ATGGAACAAAGCTGGGCAGAGAATGACTTTGACGAGTTGAGAGAGGAAGGCTTCAGAAGA
TCAAACTACTCCAAGCTAAAGGAGGAAGTTCGAACAAACGGCAAAGAAGTAAAAAACTTT
GAAAAAAAATTAGATGAATGGATAACTAGAATAACCAATGCACAGAAGTCCTTAAAGGAC
CTGATGGAGCTGAAAACCAAGGCAGGAGAACTACGTGACAAATACACAAGCCTCAGTAAC
CGATGA
>FGENESH:[exon] Gene:  7 Exon:  1  Pos: 68183  -  68428   246 bp., chain +
ATGGAACAAAGCTGGGCAGAGAATGACTTTGACGAGTTGAGAGAGGAAGGCTTCAGAAGA
TCAAACTACTCCAAGCTAAAGGAGGAAGTTCGAACAAACGGCAAAGAAGTAAAAAACTTT
GAAAAAAAATTAGATGAATGGATAACTAGAATAACCAATGCACAGAAGTCCTTAAAGGAC
CTGATGGAGCTGAAAACCAAGGCAGGAGAACTACGTGACAAATACACAAGCCTCAGTAAC
CGATGA
>FGENESH:   7   1 exon (s)  68183  -  68428    81 aa, chain +
MEQSWAENDFDELREEGFRRSNYSKLKEEVRTNGKEVKNFEKKLDEWITRITNAQKSLKD
LMELKTKAGELRDKYTSLSNR
>FGENESH:[mRNA]   8   1 exon (s)  69467  -  70072   606 bp, chain +
ATGGCAAAGGGATCTATTCAAGAAGAAGAACTAACTATACTAAATATATATGCACCCAAT
ACAGGAGCACCCAGATTCATAAAACAAGTCCTGAGTGACCTACAAAGAGACTTAGATGCC
CACACAATAATAATGGGAGACTTTAACACCCCACTGTCAACATTAGACAGATCAACGAGA
CAGAAAGTTAACAAGGATATCCAGGAATTGGACTCAGCTCTGCACCAAGCAGACCTAATA
GACATCTACAGAACTCTCCACCCCAAATCAACAGAATATACATTCTTTTCAGCACCACAC
CACACCTATTCCAAAACTGACCACATAGTTGGAAGTAAAGCTCTCCTCAGCAAATGTAAA
AGAACAGAAACTATAACAAACTGTCTCTCAGACCACAGTGCAATCAAACTAGAACTCAGG
ATTAAGAAACTCACTCAAAACCACTCAGCTACATGGAAACTGAACAGCCTGCTCCTGAAT
GACTACTGGGTACATAACAAAATGAAGGCAGAAATAAAGATGTTCTTTGAAACAACGAGA
ACAAAGACACAACACACCAGAATCTCTGAGACACATTCAAAGCAGTGTGTAGAGGGAAAT
TTATAG
>FGENESH:[exon] Gene:  8 Exon:  1  Pos: 69467  -  70072   606 bp., chain +
ATGGCAAAGGGATCTATTCAAGAAGAAGAACTAACTATACTAAATATATATGCACCCAAT
ACAGGAGCACCCAGATTCATAAAACAAGTCCTGAGTGACCTACAAAGAGACTTAGATGCC
CACACAATAATAATGGGAGACTTTAACACCCCACTGTCAACATTAGACAGATCAACGAGA
CAGAAAGTTAACAAGGATATCCAGGAATTGGACTCAGCTCTGCACCAAGCAGACCTAATA
GACATCTACAGAACTCTCCACCCCAAATCAACAGAATATACATTCTTTTCAGCACCACAC
CACACCTATTCCAAAACTGACCACATAGTTGGAAGTAAAGCTCTCCTCAGCAAATGTAAA
AGAACAGAAACTATAACAAACTGTCTCTCAGACCACAGTGCAATCAAACTAGAACTCAGG
ATTAAGAAACTCACTCAAAACCACTCAGCTACATGGAAACTGAACAGCCTGCTCCTGAAT
GACTACTGGGTACATAACAAAATGAAGGCAGAAATAAAGATGTTCTTTGAAACAACGAGA
ACAAAGACACAACACACCAGAATCTCTGAGACACATTCAAAGCAGTGTGTAGAGGGAAAT
TTATAG
>FGENESH:   8   1 exon (s)  69467  -  70072   201 aa, chain +
MAKGSIQEEELTILNIYAPNTGAPRFIKQVLSDLQRDLDAHTIIMGDFNTPLSTLDRSTR
QKVNKDIQELDSALHQADLIDIYRTLHPKSTEYTFFSAPHHTYSKTDHIVGSKALLSKCK
RTETITNCLSDHSAIKLELRIKKLTQNHSATWKLNSLLLNDYWVHNKMKAEIKMFFETTR
TKTQHTRISETHSKQCVEGNL
>FGENESH:[mRNA]   9   1 exon (s)  70355  -  70819   465 bp, chain +
ATGACACGGGGTATCACCACTGATCCCACAGAAATACAAACTACCGTCAGAGAATACTAT
AAACACCTCTACGCAAATAAACTAGAAAATCTAGAAGAAATGGATAAATTCCTCGACACA
TACACTCTGCCAAGACTAAACCAGGAAGAAGTTGTATCTCTGAATAGACCAATAACAGGC
TCTGAAATTGAGGCAATAATTAATAGCTTATCAACCAAAAAAAGTCCGGGACCAGTAGGA
TTCATAGCCGAATTCTACCAGAGGTACAAGGAGGAGCTGGTACCATTCCTTCTGAAACTA
TTCCAATCAATAGAAAAAGAGGGAATCCTCCCTAACTCATTTTATGAGGCCAGCATCATC
CTGATACCAAAGCCTGACAGAGACACAACAAAAAAAGAGAATGTTACACCAATATCCTTG
ATGAACATCGATGCAAAAATCCTCAATAAAATACTGGCAAACTGA
>FGENESH:[exon] Gene:  9 Exon:  1  Pos: 70355  -  70819   465 bp., chain +
ATGACACGGGGTATCACCACTGATCCCACAGAAATACAAACTACCGTCAGAGAATACTAT
AAACACCTCTACGCAAATAAACTAGAAAATCTAGAAGAAATGGATAAATTCCTCGACACA
TACACTCTGCCAAGACTAAACCAGGAAGAAGTTGTATCTCTGAATAGACCAATAACAGGC
TCTGAAATTGAGGCAATAATTAATAGCTTATCAACCAAAAAAAGTCCGGGACCAGTAGGA
TTCATAGCCGAATTCTACCAGAGGTACAAGGAGGAGCTGGTACCATTCCTTCTGAAACTA
TTCCAATCAATAGAAAAAGAGGGAATCCTCCCTAACTCATTTTATGAGGCCAGCATCATC
```

```
CTGATACCAAAGCCTGACAGAGACACAACAAAAAAAGAGAATGTTACACCAATATCCTTG
ATGAACATCGATGCAAAAATCCTCAATAAAATACTGGCAAACTGA
>FGENESH:   9   1 exon (s)  70355  -  70819   154 aa, chain +
MTRGITTDPTEIQTTVREYYKHLYANKLENLEEMDKFLDTYTLPRLNQEEVVSLNRPITG
SEIEAIINSLSTKKSPGPVGFIAEFYQRYKEELVPFLLKLFQSIEKEGILPNSFYEASII
LIPKPDRDTTKKENVTPISLMNIDAKILNKILAN
>FGENESH:[mRNA]  10   1 exon (s)  72135  -  72395   261 bp, chain +
ATGGGCAAGGACTTCATGTCTAAAACACCAAAACGAATGGCAACAAAAGACAAAATGGAC
AAACGGGATCTAATTAAACTAAAGAGCTTCTGCACAGCTAAAGAAACTACCATCAGAGTG
AACAGGCAACCTACAAAATGGGAGAAAATTTTTGCAATCTACTCATCTGACAAAGGGCTA
ATATCCAGAATCTACAATGAACTCAAACAAATTTACAAGAAAAAACAAACAACCCCATCA
AAAAGTGGGCAAAGGATATGA
>FGENESH:[exon] Gene: 10 Exon:  1  Pos: 72135  -  72395   261 bp., chain +
ATGGGCAAGGACTTCATGTCTAAAACACCAAAACGAATGGCAACAAAAGACAAAATGGAC
AAACGGGATCTAATTAAACTAAAGAGCTTCTGCACAGCTAAAGAAACTACCATCAGAGTG
AACAGGCAACCTACAAAATGGGAGAAAATTTTTGCAATCTACTCATCTGACAAAGGGCTA
ATATCCAGAATCTACAATGAACTCAAACAAATTTACAAGAAAAAACAAACAACCCCATCA
AAAAGTGGGCAAAGGATATGA
>FGENESH:  10   1 exon (s)  72135  -  72395    86 aa, chain +
MGKDFMSKTPKRMATKDKMDKRDLIKLKSFCTAKETTIRVNRQPTKWEKIFAIYSSDKGL
ISRIYNELKQIYKKKQTTPSKSGQRI
```

Where:

G - predicted gene number, starting from start of sequence;

Str - DNA strand (+ for direct or - for complementary);

Feature - Type (feature of coding sequence): CDSf - first (starting with start codon), CDSi - internal (internal exon), CDSl - last (ending with stop codon) coding segment, CDSo - gene contains the ONE coding exon only;

Start and **End** - Position of the Feature;

Score - Log likelihood*10 score for the feature;

ORF - start/end positions where the first codon starts and the last codon ends.

Len - length of the coding segment.

PolA - poly(A) site

Parameters:

Input	
Organism	Parameter file for specified organizm.
Sequences	Source file with nucleotide sequences in FASTA format.
Output	
Result	Name of the output file.
Print mRNA	Enabling this option results in output the nucleotide sequences of all predicted exons separately.
Print Exons	Enabling this option results in output the nucleotide sequences of all predicted exons separately.
Options	
Use GC donor splice sites:	Use GC donor splice sites: ○ **Use all potential GC sites** - Use all potential GC donor sites. ○ **Set Threshold** - Use potential GC donor splice sites with score higher the current value only.
Set Search Range	Set Search Range: ○ **Starting Position** - Set the starting position for search region in sequence.When this option is not checked, the programs uses the first nucleotide as starting one. ○ **Ending Position** - Set the ending position for search region in sequence.
Alternative Variants	Alternative Variants Output ○ **Output Variants Number** - Set the maximal number of best alternative

Output:	prediction variants to output. 🕐 **Variants Skipping Threshold** - Set the scoring threshold for the program to skip variants of prediction with score lower than the set portion of the best prediction score. I.e. if the value is set to 0.75, and the best prediction score is 1000, then all variants with score lower than 750 will be ignored. 🕐 **Number of Best Exons to Include** - Force the program to include in alternative prediction variants the set number of best exons, which were not initially included in the best prediction, sequentially. This means the program makes a prediction with the best score, after which some potential exons with high score remain unincluded in this prediction. Enabling this options forces the program to generate alternative variants that must contain the set number of these exons. 🕐 **Number of Best Sites to Include** - Force the program to include in alternative prediction variants the set number of exons with good splicing sites, which were not initially included in the best prediction, sequentially. This means the program makes a prediction with the best score, after which some potential exons with good splicing sites remain unincluded in this prediction. Enabling this options forces the program to generate alternative variants that must contain the set number of these exons. 🕐 **Stop Exons Skipping** - By default the program makes the best prediction and then tries to generate alternative variants sequentially skipping the exons, which were included in this prediction. Enabling this option prevents using this method.
Allow to Skip Promotors	During the check, for each potential promoter two alternative variants are considered: 1. The promoter is included in gene structure with formation the following 5'UTR upstream the CDS; 2. The promoter is not considered in gene structure, and predicted sequence begins directly with CDS (1st exon). Enabling this option allows both variants with following choosing of the best prediction.
Allow to Skip Terminators	During the check, for each potential terminator two alternative variants are considered: 1. The terminator is included in gene structure with formation the previous 3'UTR downstream the CDS; 2. The terminator is not considered in gene structure, and predicted sequence ends directly with CDS (last exon). Enabling this option allows both variants with following choosing of the best prediction.
Exons Restrictions	Exons Restrictions: 🕐 **First Exon Minimum** - Set the minimal allowed length for the first exon. 🕐 **Internal Exon Minimum** - Set the minimal allowed length for the internal exon. 🕐 **Single Exon Minimum** - Set the minimal allowed length for the single exon. 🕐 **Terminal Exon Minimum** - Set the minimal allowed length for the terminal exon. 🕐 **Exons Skipping Threshold** - Set the scoring threshold for the program to skip potential exons with score lower than the current one.
Specificity Factor	Set the specificity of algorithm (from -10 (High) to +10 (Low)). Increasing the parameter value results in increased number of predicted "True" exons, but the number of predicted "False" exons is also being increased. Generally, increasing of false exons prediction is drastically greater than

	increasing of true ones. Decreasing the parameter value results in symmetric situation with decreasing of predictions number.

Fgenesh+

Program for predicting multiple genes in genomic DNA sequences using HMM gene model plus homology with known protein.

Fgenesh+ was developed to analyse sequences from human, drosophila, nematode and plant, as well related organisms. The program can be used if you know protein sequence similar to protein which is predicted for a gene in your sequence. First, run any ab initio gene finding program such as Fgenes or Fgenesh. Then, run BLASTP DB search with each predicted exon. Any true predicted exon can provide you with known similar proteins, if such proteins exist in the DB. Take sequence of homologous protein and run Fgenesh+. The accuracy of gene prediction can be up to 100% depending of how similar the predicted and DB protein are.

Softberry significantly improved its gene prediction with protein support programs. New Prot_map program can be used to generate a set of gene in new organism and use them to learn parameters for gene prediction programs fgenesh and Fgenesh+. It is very useful to find pseudogenes by selection corrupted genes generated by mapping known proteins.

Speed of processing sequences

	Fgenesh+	Prot_map	GeneWise
88 sequences of genes < 20 kb	~1 min	~1 min	~90 min
8 sequences of genes > 400000 kb	~1 min	~1 min	~1200 min

Prot_map mapping of Human protein set of 55946 proteins on chromosome 19 (~59 MB) takes just 90 min (best hit for each protein) and 148 min (all significant hits for each protein).

Accuracy comparison

Comparison of accuracy of gene prediction by ab initio Fgenesh and prediction with protein support by Fgenesh+ or GenWise and Prot_map - mapping protein to human DNA is done on large set of human genes with using mouse or drosophila homologous proteins. We can see that Fgenesh+ shows the best performance with mouse proteins. With Drosophila proteins ab initio prediction Fgenesh works better than GeneWise for all ranges of similarity and Fgenesh+ is the best predictor if similarity is higher 60%.

Gene prediction with mouse protein support:
Similarity level > 90% - 921 sequences

	Sn ex	Sno ex	Sp ex	Sn nuc	Sp nuc	CC	%CG
Fgenesh	86.2	91.7	88.6	93.9	93.4	0.9334	34
Genwise	93.9	97.6	95.9	99.0	99.6	0.9926	66
Fgenesh+	97.3	98.9	98.0	99.1	99.6	0.9936	81
Prot_map	95.9	98.3	96.9	99.1	99.5	0.9924	73

Gene prediction with Drosophila proteins with similarity ranging from 22% to 98% and coverage in both proteins > 75%:

1. Similarity level > 80% - 66 sequences.

	Sn ex	Sno ex	Sp ex	Sn nuc	Sp nuc	CC	%CG
Fgenesh	90.5	93.8	95.1	97.9	96.9	0.950	55
Genwise	79.3	83.9	86.8	97.3	99.5	0.985	23
Fgenesh+	95.1	97.8	97.0	98.9	99.5	0.9914	70
Prot_map	86.4	95.3	88.1	97.6	99.0	0.982	41

Ab initio gene prediction programs usually correctly predict significant fraction of exons in a gene, but they often assemble gene in incorrect way: combine several genes or split one gene into several, skip exons or include false exons. Using similarity information provided by one or several true predicted exons can significantly improve accuracy of gene finding.

You should provide similarity value known from the Blast or Prot_map search - it affects prediction. The programs uses similarity to estimate how similar the predicted gene product can be from its homolog.

To use the program, click (mark) Human, Drosophila, Nematode or Plant button and FGENESH button. Paste your sequence to the first window or load your file with nucleotide sequence in FASTA format. Paste your protein sequence to the second window.

Fgenesh+ output:

G - predicted gene number, starting from start of sequence; Str - DNA strand (+ for direct or - for complementary);
Feature - type of coding sequence: CDSf - First (Starting with Start codon), CDSi - internal (internal exon), CDSl - last coding segment, ending with stop codon);
TSS - Position of transcription start (TATA-box position and score);
Start and End - Position of the Feature;
Weight - Log likelihood*10 score for the feature ORF - start/end positions where the first complete codon starts and the last codon ends Last three values: Length of exon, positions in protein, percent of similarity with target protein

```
  FGENESH+ 2.5 Prediction of potential genes in Homo_sapiens genomic DNA
 Time    :   Sun Jan 28 22:28:20 2007
 Seq name: >Adh_and_cact.1 (2919020 bases) 848501 853000
 Length of sequence: 4500
 Homology: gi|2313041|gnl|PID|d1022564 (D84316) rab14 [Drosophila
melanogaster]
 Length of homolog: 215
 Number of predicted genes 1 in +chain 1 in -chain 0
 Number of predicted exons 4 in +chain 4 in -chain 0
 Positions of predicted genes and exons: Variant   1 from   1,
Score:1130.648633
   G Str   Feature   Start        End    Score           ORF           Len

   1 +      TSS       1459             -9.69
   1 +    1 CDSf      2585 -      2690  190.55      2585 -      2689    105     1      35  100
   1 +    2 CDSi      2756 -      2936  334.25      2758 -      2934    177    37      95  100
   1 +    3 CDSi      2991 -      3173  315.47      2992 -      3171    180    97     156  100
   1 +    4 CDSl      3242 -      3419  302.12      3243 -      3419    177   158     214  100
   1 +      PolA      3968              1.13

Predicted protein(s):
>FGENESH:   1   4 exon (s)   2585  -   3419   215 aa, chain +
MTAAPYNYNYIFKYIIIGDMGVGKSCLLHQFTEKKFMANCPHTIGVEFGTRIIEVDDKKI
KLQIWDTAGQERFRAVTRSYYRGAAGALMVYDITRRSTYNHLSSWLTDTRNLTNPSTVIF
LIGNKSDLESTREVTYEEAKEFADENGLMFLEASAMTGQNVEEAFLETARKIYQNIQEGR
LDLNASESGVQHRPSQPSRTSLSSEATGAKDQCSC
```

Parameters:

Input	
Sequences	Set your source file with nucleotide sequences in FASTA format.
Homologous Sequence(s)	Set your source file with homologous sequences in FASTA format.
Organism	Parameter file for specified organizm.

Output	
Result	Name of the output file.
Print mRNA	Enabling this option results in output the nucleotide sequences of all predicted exons separately.
Print Exons	Enabling this option results in output the nucleotide sequences of all predicted exons separately.
Threshold for Flanking Exons	This option specifies the minimal allowed length for flanking exons, which has no similarity with homologous sequence, to output.
Options	
Minimal Exon Homology	Exon is considered as completely unsimilar, if its similarity with the homologue is less than the value specified (in percents).
Costs for Exons Homology:	Costs for Exons Homology: ◷ **Exons Homology Bonus** - If a potential exon has a similarity with given homolog, its resulting score will be equal to intial score plus the score of homology multiplied by the set value. ◷ **Penalty for Non-Homologous Exons** - This option specifies a penalty for the internal predicted exons, which have no similarity to homologue and lie between the exons possessing homology.
Use GC donor splice sites:	Use GC donor splice sites: ◷ **Use all potential GC sites** - Use all potential GC donor sites. ◷ **Set Threshold** - Use potential GC donor splice sites with score higher the current value only.
Set Search Range	Set Search Range: ◷ **Starting Position** - Set the starting position for search region in sequence.When this option is not checked, the programs uses the first nucleotide as starting one. ◷ **Ending Position** - Set the ending position for search region in sequence.
Alternative Variants Output:	Alternative Variants Output ◷ **Output Variants Number** - Set the maximal number of best alternative prediction variants to output. ◷ **Variants Skipping Threshold** - Set the scoring threshold for the program to skip variants of prediction with score lower than the set portion of the best prediction score. I.e. if the value is set to 0.75, and the best prediction score is 1000, then all variants with score lower than 750 will be ignored. ◷ **Number of Best Exons to Include** - Force the program to include in alternative prediction variants the set number of best exons, which were not initially included in the best prediction, sequentially. This means the program makes a prediction with the best score, after which some potential exons with high score remain unincluded in this prediction. Enabling this options forces the program to generate alternative variants that must contain the set number of these exons. ◷ **Number of Best Sites to Include** - Force the program to include in alternative prediction variants the set number of exons with good splicing sites, which were not initially included in the best prediction, sequentially. This means the program makes a prediction with the best score, after which some potential exons with good splicing sites remain unincluded in this prediction. Enabling this options forces the program to generate alternative variants that must contain the set number of these exons. ◷ **Stop Exons Skipping** - By default the program makes the best prediction and then tries to generate alternative variants sequentially skipping the exons, which were included in this prediction. Enabling this option prevents using this

	method.
Allow to Skip Promotors	During the check, for each potential promoter two alternative variants are considered: 1. The promoter is included in gene structure with formation the following 5'UTR upstream the CDS; 2. The promoter is not considered in gene structure, and predicted sequence begins directly with CDS (1st exon). Enabling this option allows both variants with following choosing of the best prediction.
Allow to Skip Terminators	During the check, for each potential terminator two alternative variants are considered: 1. The terminator is included in gene structure with formation the previous 3'UTR downstream the CDS; 2. The terminator is not considered in gene structure, and predicted sequence ends directly with CDS (last exon). Enabling this option allows both variants with following choosing of the best prediction.
Exons Restrictions	Exons Restrictions: 🕐 **First Exon Minimum** - Set the minimal allowed length for the first exon. 🕐 **Internal Exon Minimum** - Set the minimal allowed length for the internal exon. 🕐 **Single Exon Minimum** - Set the minimal allowed length for the single exon. 🕐 **Terminal Exon Minimum** - Set the minimal allowed length for the terminal exon. 🕐 **Exons Skipping Threshold** - Set the scoring threshold for the program to skip potential exons with score lower than the current one.
Specificity Factor	Set the specificity of algorithm (from -10 (High) to +10 (Low)). Increasing the parameter value results in increased number of predicted "True" exons, but the number of predicted "False" exons is also being increased. Generally, increasing of false exons prediction is drastically greater than increasing of true ones. Decreasing the parameter value results in symmetric situation with decreasing of predictions number.

Fgenesh-2

Program for predicting multiple genes in genomic DNA sequences using HMM gene model and genomic sequences of two close organisms to increase reliability of true exon and gene identification

The program can be used if DNA sequences of homologous genomic regions of two similar organisms, such as Human and mouse, are available.

Ab initio gene prediction programs usually correctly predict significant fraction of exons in a gene, but they often assemble gene in incorrect way: combine several genes or split one gene into several, skip exons or include false exons. Using sequences of two organisms can significantly improve accuracy of EXACT gene finding, taking into accunt that Human genome draft sequence and Mouse genomic sequence provide a lot of homologous sequences.

Program shows predicted genes in both sequences as two sequential Fgenesh outputs.

G - predicted gene number, starting from start of sequence; Str - DNA strand (+ for direct or - for complementary);

Feature - type of coding sequence: CDSf - First (Starting with Start codon), CDSi - internal (internal exon), CDSl - last coding segment, ending with stop codon);

TSS - Position of transcription start (TATA-box position and score);

Start and End - Position of the Feature;

Weight - Log likelihood*10 score for the feature ORF - start/end positions where the first complete codon starts and the last codon ends Last three values: Length of exon, positions in protein, percent of similarity with target protein

EXAMPLE of output for genes predicted in Human and Mouse genomic sequences:

```
 Fgenesh-2 1.C Prediction of potential genes in 1st genomic DNA
 Time:   Fri Nov 10 02:55:51 2000
 Seq name: HSCKIIBE
 Length of sequence:  5917  GC content: 53 Zone: 3
 Number of predicted genes 1 in +chain 1 in -chain 0
 Number of predicted exons 6 in +chain 6 in -chain 0
 Positions of predicted genes and exons:
  G Str Feature    Start     End   Score        ORF           Len

  1 +   1 CDSf    1634 -    1705     18.99    1634 -    1705     72
  1 +   2 CDSi    2672 -    2774     38.26    2672 -    2773    102
  1 +   3 CDSi    3344 -    3459     41.09    3346 -    3459    114
  1 +   4 CDSi    3906 -    3981     25.73    3906 -    3980     75
  1 +   5 CDSi    4128 -    4317     67.44    4130 -    4315    186
  1 +   6 CDSl    4645 -    4735     29.35    4646 -    4735     90
  1 +     PolA    4855                0.92

Predicted protein(s):
>Fgenesh-2   1   6 exon (s)   1634  -   4735    215 aa, chain +
MSSSEEVSWISWFCGLRGNEFFCEVDEDYIQDKFNLTGLNEQVPHYRQALDMILDLEPDE
ELEDNPNQSDLIEQAAEMLYGLIHARYILTNRGIAQMLEKYQQGDFGYCPRVYCENQPML
PIGLSDIPGEAMVKLYCPKCMDVYTPKSSRHHHTDGAYFGTGFPHMLFMVHPEYRPKRPA
NQFVPRLYGFKIHPMAYQLQLQAASNFKSPVKTIR
 Fgenesh-2 1.C Prediction of potential genes in 2nd genomic DNA
 Time:   Fri Nov 10 02:55:51 2000
 Seq name: MMGMCK2B
 Length of sequence:  7874  GC content: 51 Zone: 2
 Number of predicted genes 1 in +chain 1 in -chain 0
 Number of predicted exons 6 in +chain 6 in -chain 0
 Positions of predicted genes and exons:
  G Str Feature    Start     End   Score        ORF           Len

  1 +   1 CDSf    2169 -    2240     38.64    2169 -    2240     72
  1 +   2 CDSi    2829 -    2931     28.70    2829 -    2930    102
  1 +   3 CDSi    4112 -    4227     36.45    4114 -    4227    114
  1 +   4 CDSi    4615 -    4690     18.76    4615 -    4689     75
  1 +   5 CDSi    4801 -    4990     56.00    4803 -    4988    186
  1 +   6 CDSl    6262 -    6352     18.70    6263 -    6352     90
  1 +     PolA    6470                0.92

Predicted protein(s):
>Fgenesh-2   1   6 exon (s)   2169  -   6352    215 aa, chain +
MSSSEEVSWISWFCGLRGNEFFCEVDEDYIQDKFNLTGLNEQVPHYRQALDMILDLEPDE
ELEDNPNQSDLIEQAAEMLYGLIHARYILTNRGIAQMLEKYQQGDFGYCPRVYCENQPML
PIGLSDIPGEAMVKLYCPKCMDVYTPKSSRHHHTDGAYFGTGFPHMLFMVHPEYRPKRPA
NQFVPRLYGFKIHPMAYQLQLQAASNFKSPVKTIR
```

Parameters:

Input	
Organism	Parameter file for specified organizm.
Sequences	Source file with nucleotide sequences in FASTA format.
File	Source file with second nucleotide sequence in FASTA format.
Output	
Result	Name of the output file.
Options	

Protein similarity	Write % of protein similarity you expect.

Fgenesh-c

Program for predicting multiple genes in genomic DNA sequences using HMM gene model plus similarity with known mRNA/EST

The program can be used if you know mRNA/EST sequence that is homologous to that of predicted gene. First, run any ab initio gene finding program such as Fgenes or Fgenesh. Then, run BLAST DB search with each predicted exon. If homologous mRNA is found, use it to improve accuracy of assembly of your predicted gene.

Ab initio gene prediction programs usually correctly predict significant fraction of exons in a gene, but they often assemble gene in incorrect way: combine several genes or split one gene into several, skip exons or include false exons. Using mRNA homology information provided by one or several true predicted exons can significantly improve accuracy of gene finding.

Program use and output are similar to those of Fgenesh+:
G - predicted gene number, starting from start of sequence;
Str - DNA strand (+ for direct or - for complementary);
Feature - type of coding sequence: CDSf - First (Starting with Start codon), CDSi - internal (internal exon), CDSl - last coding segment, ending with stop codon);
TSS - Position of transcription start (TATA-box position and score);
Start and End - Position of the Feature;
Weight - Log likelihood*10 score for the feature ORF - start/end positions where the first complete codon starts and the last codon ends Last three values: Length of exon, positions in protein, percent of similarity with target protein

Output example:

```
 FGENESHc 2.5 Prediction of potential genes in Homo_sapiens genomic DNA
 Time    :   Sun Jan 28 23:16:55 2007
 Seq name: >HUMSFRS_8213_DNA_14-FEB-1996
 Length of sequence: 6423
 Homology: Q
 Length of homolog: 817
 Number of predicted genes 1 in +chain 1 in -chain 0
 Number of predicted exons 8 in +chain 8 in -chain 0
 Positions of predicted genes and exons: Variant   1 from   1,
Score:437.471680
   G Str    Feature    Start         End     Score           ORF            Len

   1 +      TSS        16              -7.39
   1 +    1 CDSf       151 -       178   59.16      151 -       177     27      1      78  100
   1 +    2 CDSi      1213 -      1393  118.23     1215 -      1391    177     79     259  100
   1 +    3 CDSi      1702 -      1878   97.79     1703 -      1876    174    260     436  100
   1 +    4 CDSi      2754 -      2828   40.58     2755 -      2826     72    437     511  100
   1 +    5 CDSi      3250 -      3360   38.73     3251 -      3358    108    512     622  100
   1 +    6 CDSi      4659 -      4712   23.03     4660 -      4710     51    623     676  100
   1 +    7 CDSi      5227 -      5262   24.08     5228 -      5260     33    677     712  100
   1 +    8 CDSl      6219 -      6273   52.07     6220 -      6273     54    713     817  100
   1 +      PolA      6378             -6.78

Predicted protein(s):
>FGENESH:   1   8 exon (s)    151  -   6273   238 aa, chain +
MSRYGRYGGETKVYVGNLGTGAGKGELERAFSYYGPLRTVWIARNPPGFAFVEFEDPRDA
EDAVRGLDGKVICGSRVRVELSTGMPRRSRFDRPPARRPFDPNDRCYECGEKGHYAYDCH
RYSRRRRSRSRSRSHSRSRGRRYSRSRSRSRGRRSRSASPRRSRSISLRRSRSASLRRSR
SGSIKGSRYFQSPSRSRSRSRSISRPRSSRSKSRSPSPKRSRSPSGSPRRSASPERMD
```

Parameters:

Input	
Organism	Select parameter file for specified organizm.
Sequences	Set your source file with nucleotide sequences in FASTA format.
Homologous Sequence(s)	Set your source file with cDNA/EST in FASTA format.
Output	
Result	Name of the output file.
Print mRNA	Enabling this option results in output the nucleotide sequences of all predicted exons separately.
Print Exons	Enabling this option results in output the nucleotide sequences of all predicted exons separately.
Threshold for Flanking Exons	This option specifies the minimal allowed length for flanking exons, which has no similarity with homologous sequence, to output.
Options	
Minimal Exon Homology	Exon is considered as completely unsimilar, if its similarity with the homologue is less than the value specified (in percents).
Costs for Exons Homology	If a potential exon has a similarity with given homolog, its resulting score will be equal to intial score plus the score of homology multiplied by the set value.
Costs for Exons Homology:	Costs for Exons Homology: ○ **Exons Homology Bonus** - If a potential exon has a similarity with given homolog, its resulting score will be equal to intial score plus the score of homology multiplied by the set value. ○ **Penalty for Non-Homologous Exons** - This option specifies a penalty for the internal predicted exons, which have no similarity to homologue and lie between the exons possessing homology.
Use GC donor splice sites:	Use GC donor splice sites: ○ **Use all potential GC sites** - Use all potential GC donor sites. ○ **Set Threshold** - Use potential GC donor splice sites with score higher the current value only.
Set Search Range	Set Search Range: ○ **Starting Position** - Set the starting position for search region in sequence.When this option is not checked, the programs uses the first nucleotide as starting one. ○ **Ending Position** - Set the ending position for search region in sequence.
Alternative Variants Output:	Alternative Variants Output ○ **Output Variants Number** - Set the maximal number of best alternative prediction variants to output. ○ **Variants Skipping Threshold** - Set the scoring threshold for the program to skip variants of prediction with score lower than the set portion of the best prediction score. I.e. if the value is set to 0.75, and the best prediction score is 1000, then all variants with score lower than 750 will be ignored. ○ **Number of Best Exons to Include** - Force the program to include in alternative prediction variants the set number of best exons, which were not initially included in the best prediction, sequentially. This means the program makes a prediction with the best score, after which some potential exons with high score remain unincluded in this prediction. Enabling this options forces the program to generate alternative variants that must contain the set number of these exons. ○ **Number of Best Sites to Include** - Force the program to include in

	alternative prediction variants the set number of exons with good splicing sites, which were not initially included in the best prediction, sequentially. This means the program makes a prediction with the best score, after which some potential exons with good splicing sites remain unincluded in this prediction. Enabling this options forces the program to generate alternative variants that must contain the set number of these exons. 🕐 **Stop Exons Skipping** - By default the program makes the best prediction and then tries to generate alternative variants sequentially skipping the exons, which were included in this prediction. Enabling this option prevents using this method.
Allow to Skip Promotors	During the check, for each potential promoter two alternative variants are considered: 1. The promoter is included in gene structure with formation the following 5'UTR upstream the CDS; 2. The promoter is not considered in gene structure, and predicted sequence begins directly with CDS (1st exon). Enabling this option allows both variants with following choosing of the best prediction.
Allow to Skip Terminators	During the check, for each potential terminator two alternative variants are considered: 1. The terminator is included in gene structure with formation the previous 3'UTR downstream the CDS; 2. The terminator is not considered in gene structure, and predicted sequence ends directly with CDS (last exon). Enabling this option allows both variants with following choosing of the best prediction.
Exons Restrictions	Exons Restrictions: 🕐 **First Exon Minimum** - Set the minimal allowed length for the first exon. 🕐 **Internal Exon Minimum** - Set the minimal allowed length for the internal exon. 🕐 **Single Exon Minimum** - Set the minimal allowed length for the single exon. 🕐 **Terminal Exon Minimum** - Set the minimal allowed length for the terminal exon. 🕐 **Exons Skipping Threshold** - Set the scoring threshold for the program to skip potential exons with score lower than the current one.
Specificity Factor	Set the specificity of algorithm (from -10 (High) to +10 (Low)). Increasing the parameter value results in increased number of predicted "True" exons, but the number of predicted "False" exons is also being increased. Generally, increasing of false exons prediction is drastically greater than increasing of true ones. Decreasing the parameter value results in symmetric situation with decreasing of predictions number.

Fgenesh_2_gff3

Fgenesh_2_gff3 utilizes the results of FGenesh and represents them in gff3 format (http://www.sequenceontology.org/resources/gff3.html).

Parameters:

Input file	Browse your source file with Fgenesh result.
Output file	Name of the output file.

FSplice

Program provides the possibility to search for both donor and acceptor sites, and to define thresholds for them independently. Program allows to search minor variants of splicing donor site (GC-site) as well.

Output example

```
FSplice 1.0. Prediction of potential splice sites in Homo_sapiens genomic DNA
 Seq name: NM_000449 chr 1 - 148089557 148094091 4535
 Length of sequence: 4535
Direct  chain.
 Acceptor(AG) sites. Treshold      4.175 (90%).
       1 P:     187 W:  7.47 Seq: attctAGccctc
       2 P:     296 W:  6.42 Seq: tcttcAGaggct
       3 P:     495 W:  7.30 Seq: tccctAGcagtc
       4 P:     498 W:  5.72 Seq: ctagcAGtcaga
       5 P:     559 W: 14.18 Seq: cccacAGcaagg
       6 P:     847 W:  6.42 Seq: atggtAGcctat
       7 P:    1332 W:  9.70 Seq: acctcAGcaaga
       8 P:    1383 W:  9.25 Seq: ccttcAGctccc
       9 P:    1393 W:  5.38 Seq: ccctcAGgaccc
      10 P:    1673 W:  9.95 Seq: tctgtAGctcag
      11 P:    1721 W:  4.72 Seq: cctatAGgtgga
      12 P:    1916 W:  6.72 Seq: tccctAGggact
      13 P:    1984 W:  9.70 Seq: cactcAGgaagt
      14 P:    2366 W: 12.18 Seq: ctcccAGgtaaa
      15 P:    2467 W:  7.12 Seq: cctgtAGctgag
      16 P:    2638 W:  7.42 Seq: acttcAGccaga
      17 P:    2779 W:  6.42 Seq: gctacAGcagca
      18 P:    2867 W:  6.42 Seq: gtctcAGcaacc
      19 P:    2995 W:  5.03 Seq: ctaccAGtcagt
      20 P:    3033 W:  5.85 Seq: tcctcAGtttcc
      21 P:    3078 W:  9.68 Seq: tctgcAGaagag
      22 P:    3342 W:  9.88 Seq: tttttAGcctcc
      23 P:    3545 W:  8.12 Seq: cccccAGgcttt
      24 P:    4435 W:  6.70 Seq: tcctaAGgaagt
      25 P:    4458 W:  6.65 Seq: tgtacAGacagc
      26 P:    4513 W:  5.65 Seq: ttttcAGcttga
      27 P:    4533 W:  4.58 Seq: gctttAGtg---
 Donor(GT) sites. Treshold      6.099 (90%).
       1 P:      40 W:  8.20 Seq: aagtgGTgagaa
       2 P:     150 W:  7.50 Seq: ccagtGTgagtt
       3 P:     307 W:  7.64 Seq: ccgagGTaccat
       4 P:     317 W:  9.32 Seq: atttcGTaagta
       5 P:     594 W: 15.48 Seq: tcctgGTaagtg
       6 P:     691 W:  9.60 Seq: gagagGTagggt
       7 P:    1416 W: 13.38 Seq: aaaagGTaggtt
       8 P:    1794 W:  7.36 Seq: tatcgGTgggtg
       9 P:    2325 W: 10.44 Seq: agagtGTaagta
      10 P:    2367 W: 13.10 Seq: cccagGTaaaag
      11 P:    2438 W:  8.06 Seq: tctagGTatgat
      12 P:    2841 W:  7.36 Seq: cgctgGTgtgtt
      13 P:    3180 W: 14.08 Seq: cccagGTaagga
      14 P:    3733 W: 10.16 Seq: gagagGTaggca
      15 P:    3796 W:  8.62 Seq: tacctGTgagtg
      16 P:    4177 W: 11.56 Seq: caaaaGTgagtg
      17 P:    4237 W:  6.38 Seq: gagagGTagaca
      18 P:    4341 W:  8.06 Seq: tacagGTctgtg
Reverse chain.
 Acceptor(AG) sites. Treshold      4.175 (90%).
       1 P:     193 W:  6.42 Seq: cccacAGacctg
       2 P:      292 W:  5.40 Seq: ggtgcAGtgtct
       3 P:      316 W:  4.58 Seq: gccaaAGgaaaa
       4 P:      481 W:  8.07 Seq: ttttcAGcctct
       5 P:      517 W: 10.38 Seq: cctccAGctgag
       6 P:      646 W:  4.17 Seq: tttcgAGggcgc
       7 P:      709 W:  7.05 Seq: gctttAGctggt
       8 P:      742 W:  6.70 Seq: ctcacAGgtact
       9 P:     1424 W:  5.67 Seq: ggtttAGatgac
      10 P:     1463 W:  6.97 Seq: tctgcAGaggta
      11 P:     1964 W:  7.45 Seq: ttgtcAGagatc
      12 P:     2035 W:  6.78 Seq: attgcAGaagcc
      13 P:     2068 W:  7.25 Seq: gcctcAGctaca
      14 P:     2287 W:  4.72 Seq: actgtAGcaata
      15 P:     2397 W:  9.20 Seq: ctcccAGgtcct
      16 P:     2421 W:  4.40 Seq: tctctAGtcaag
      17 P:     2748 W:  5.08 Seq: ccgatAGgcatc
      18 P:     2798 W:  5.47 Seq: cttccAGgtggt
      19 P:     3064 W:  6.58 Seq: ttcccAGtgaac
      20 P:     3133 W: 10.05 Seq: tctccAGtggtg
      21 P:     3901 W:  9.50 Seq: ccctcAGcattt
      22 P:     3945 W:  6.03 Seq: ttaccAGgatcc
      23 P:     4298 W:  4.72 Seq: cccccAGtcttg
      24 P:     4406 W: 11.57 Seq: tccccAGaaggc
      25 P:     4440 W:  9.12 Seq: tacccAGaaagg
 Donor(GT) sites. Treshold      6.099 (90%).
       1 P:       31 W:  8.48 Seq: aaaagGTcagag
       2 P:       49 W: 10.02 Seq: accagGTactaa
       3 P:      400 W:  7.08 Seq: ctttgGTatgct
       4 P:      743 W: 10.02 Seq: cacagGTacttc
       5 P:      832 W:  6.80 Seq: gctgaGTgagtc
       6 P:      896 W: 12.40 Seq: agttgGTaagat
       7 P:     1218 W:  7.64 Seq: acacaGTaaggt
       8 P:     1223 W:  8.90 Seq: gtaagGTgtgaa
       9 P:     1466 W:  7.64 Seq: cagagGTaccaa
      10 P:     1477 W: 12.26 Seq: aaaagGTaatag
      11 P:     1491 W: 11.84 Seq: tgaagGTgagga
      12 P:     1830 W:  7.64 Seq: cacagGTcaggg
      13 P:     2196 W:  6.94 Seq: ggaagGTgattt
      14 P:     2686 W:  6.80 Seq: catggGTgaggg
      15 P:     2982 W:  7.22 Seq: ccctgGTaaacc
      16 P:     3159 W:  9.32 Seq: tgaagGTagaga
      17 P:     3209 W: 10.16 Seq: ctgagGTaggag
      18 P:     3773 W:  6.80 Seq: atcaaGTgagag
      19 P:     4253 W:  8.34 Seq: gggtgGTaggtt
```

Where:

Acceptor(AG) sites. - the type of splicing sites. For the current case "Acceptor(AG)" means the U2-type acceptor site. Possible variants: Donor(GT) sites. means U2-type donor GT-site (Major variant). Donor(GC) sites. means U2-type donor GC- site (Minor variant).

Treshold 4.175 (90%) - means that for the current threshold value (4.175) 90% of true splicing sites are being classified as true.

P: 187 - position of splicing site

W: - weight of site.

Parameters:

Input	
Organism	Select parameter file for specified organizm.

Sequences	Set your source file with nucleotide sequences in FASTA format.
Output	
Output file	Name of output file.
Options	
Splice site sequence length	Output splice site flank's length (default value is 5).
Splice site threshold	Splice site threshold (default value is 90).
Scan target sequence in different chain	Scan target sequence in different chain: **In direct chain only (default)** **In reverse chain only** **In both chains**

PDFGenes

PDFGenes utilizes the results of Gene Finding software, such as **FGenesh**, **FGenesh+**, **FGenesh-C**, **FGenesh-2**, **FGenes**, **FGenes-m** and **BestORF**, and represents them in PDF format for better viewability.

Parameters:

Input	
File with Prediction	File with prediction from Gene Finding software. Results of the following programs can be used: **FGenesh** **FGenesh+** **FGenesh-C** **FGenesh-2** **FGenes** **FGenes-m** **BestORF**
Output	
Result	Name of output file

PSF

Finding pseudogenes in a genomic sequence.
Searching for pseudogenes is performed by aligning set of proteins with the genomic sequence. Protein FASTA-file could contain sequences with unformatted names or (preferably) with specially formatted ones. Proteins with formatted names are produced with a PSF_Pre program (not installed in the current version). This special prot. name format describes nucleotide sequence which translation gives appropriate protein, and number of its exons.

All the alignments containing one of the following are considered pseudogene candidates:
(1) stop-codons/frameshifts in nuc. sequence [for alignment with ANY protein]
(2) PolyA site and/or PolyA signal, if exon is single [for alignment with ANY protein]
(3) Number of exons is much lower than in ancestor gene [for alignment with protein SPECIALLY FORMATTED]
(4) Ka/Ks ratio exceeds 0.5 [for alignment with protein SPECIALLY FORMATTED]

It is recommended to input NR or IPI base as a protein base (better unredundant). In this case only p.(1) and p.(2) will work, but resulting candidates will be more reliable. Note that incorrectly predicted proteins might give a number of false pseudogenes.

Output example:

```
chr @@ chain @@ pos(dir.ch.) @@ len(nt.) @@ identity,@@ coverage,@@ Ka/Ks @@ uali.head
@@ uali.tail @@ exons#,lower @@ exons#,upper @@ polyA @@ polyA_signal @@ corr.stops#
@@ uncorr.stops# @@ corr.frameshifts# @@ uncorr.frameshifts# @@ prototype_chr @@
prototype_prot_name @@ prototype_exon#,lower @@ prototype_exon#,upper @@ DNA_identity
@@ CDS length
ENm009 @@ - @@ 322971 @@ 859 @@ 57.79 @@ 81.61 @@ 0.283 @@ 0 @@ 13 @@ 1 @@ 1 @@ 0 @@ 0
@@ 0 @@ 0 @@ 0 @@ 1 @@ chr11 @@ C11000184 chr11 1 exon (s) 424011 - 423106 ORF: 1 -
900  299 aa, chain - ## BY PROTMAP: gi|21928977|dbj|BAC06074.1| seven transmembrane
helix receptor [Homo ## 29 @@ 1 @@ 1 @@ 60.656 @@ 732 @@
ENm009 @@ + @@ 966139 @@ 872 @@ 49.59 @@ 75.63 @@ 0.487 @@ 10 @@ 19 @@ 1 @@ 2 @@ 0 @@
0 @@ 0 @@ 0 @@ 0 @@ 1 @@ chr11 @@ C11000197 chr11 1 exon (s) 433690 - 432722 ORF: 242
- 1204 orf 4667288 4668250  320 aa, chain - ## gi|13540539|ref|NP_110401.1|
(NM_030774) olfactory receptor, family 51, subfamily E, member 2; prostate specific G-
protein coupled receptor [Homo sapiens] ## 320 ## orf_perfect ##
NM_030774_#_242_#_1204 @@ 1 @@ 1 @@ 60.882 @@ 726 @@
ENm009 @@ + @@ 33573 @@ 928 @@ 62.29 @@ 95.19 @@ 0.284 @@ 3 @@ 1 @@ 1 @@ 1 @@ 0 @@ 0
@@ 0 @@ 0 @@ 0 @@ 1 @@ chr11 @@ C11000202 chr11 1 exon (s) 437411 - 436467 ORF: 1 -
939  312 aa, chain - ## BY PROTMAP: gi|22061831|ref|XP_171424.1| similar to olfactory
receptor [Pan trog ## 31 @@ 1 @@ 1 @@ 66.105 @@ 891 @@
....
```

Where:

Fields are separated with '@@' sequence.
First line represent field names.
List of field names:

chr	chromosome (or another sequence) name is which search has been carried out
chain	chain
pos(dir.ch.)	(nt.) pseudogene start position (in direct chain)
len(nt.)	(nt.) pseudogene length. Note thate pseudogene lies from the right of 'pos(dir.ch)'
identity	(%) Identity with a protein (0...100%).
coverage	(%) Coverage of a protein with alignment
Ka/Ks	ratio calulated by Nei-Gojobori method
uali.head	(yes/no) first codon of alignment is ATG
uali.tail	(yes/no) last codon of alignment is stop-codon
exons#,lower	number of exons, lower estimation
exons#,upper	number of exons, upper estimation
polyA	(yes/no) there is a polyA tail at the 3' terminus of alignment
polyA_signal	(yes/no) there is a polyA signal at the 3' terminus of alignment
corr.stops#	number of correctable (by one mismatch) in-frame stop codons
uncorr.stops#	number of uncorrectable (by one mismatch) in-frame stop codons
corr.frameshifts#	number of correctable (by one-nucleotide instertion/deletion) frameshifts
uncorr.frameshifts#	number of incorrectable (by one-nucleotide instertion/deletion) frameshifts
prototype_chr	chromosome of prototype protein gene
prototype_prot_name	prototype protein gene name
prototype_exon#,lower	number of exons of prototype prot. gene, lower estimation
prototype_exon#,upper	number of exons of prototype prot. gene, upper estimation
DNA_identity	Identity between prototype gene and pseudogene at the level of

	DNA
CDS length	(nt.) CDS length

Parameters:

<table>
<tr><th colspan="2">Input</th></tr>
<tr><td><b>Nucleotide sequence</b></td><td>Nucleotide FASTA-file with a single genomic sequence (without gaps).</td></tr>
<tr><td><b>Protein set</b></td><td>MultiFASTA-file with protein sequences, without gaps. Headers can include additional information in Softberry <b>AbInitio</b> or <b>FGENESH++</b> format. Here IPI or NR database could be given on input.</td></tr>
<tr><th colspan="2">Output</th></tr>
<tr><td><b>Output file</b></td><td>Specially formatted file with the pseudogenes descriptions.
Fields are separated with '@@' sequence.
List of fields:
<table>
<tr><td><b>chr</b></td><td>chromosome (or another sequence) name is which search has been carried out</td></tr>
<tr><td><b>chain</b></td><td>chain</td></tr>
<tr><td><b>pos(dir.ch.)</b></td><td>(nt.) pseudogene start position (in direct chain)</td></tr>
<tr><td><b>len(nt.)</b></td><td>(nt.) pseudogene length. Note thate pseudogene lies from the right of 'pos(dir.ch)'</td></tr>
<tr><td><b>identity</b></td><td>(%) Identity with a protein (0...100%).</td></tr>
<tr><td><b>coverage</b></td><td>(%) Coverage of a protein with alignment</td></tr>
<tr><td><b>Ka/Ks</b></td><td>ratio calulated by Nei-Gojobori method</td></tr>
<tr><td><b>uali.head</b></td><td>(yes/no) first codon of alignment is ATG</td></tr>
<tr><td><b>uali.tail</b></td><td>(yes/no) last codon of alignment is stop-codon</td></tr>
<tr><td><b>exons#,lower</b></td><td>number of exons, lower estimation</td></tr>
<tr><td><b>exons#,upper</b></td><td>number of exons, upper estimation</td></tr>
<tr><td><b>polyA</b></td><td>(yes/no) there is a polyA tail at the 3' terminus of alignment</td></tr>
<tr><td><b>polyA_signal</b></td><td>(yes/no) there is a polyA signal at the 3' terminus of alignment</td></tr>
<tr><td><b>corr.stops#</b></td><td>number of correctable (by one mismatch) in-frame stop codons</td></tr>
<tr><td><b>uncorr.stops#</b></td><td>number of uncorrectable (by one mismatch) in-frame stop codons</td></tr>
<tr><td><b>corr.frameshifts#</b></td><td>number of correctable (by one-nucleotide instertion/deletion) frameshifts</td></tr>
<tr><td><b>uncorr.frameshifts#</b></td><td>number of incorrectable (by one-nucleotide instertion/deletion) frameshifts</td></tr>
<tr><td><b>prototype_chr</b></td><td>chromosome of prototype protein gene</td></tr>
<tr><td><b>prototype_prot_name</b></td><td>prototype protein gene name</td></tr>
<tr><td><b>prototype_exon#,lower</b></td><td>number of exons of prototype prot. gene, lower estimation</td></tr>
<tr><td><b>prototype_exon#,upper</b></td><td>number of exons of prototype prot. gene, upper estimation</td></tr>
<tr><td><b>DNA_identity</b></td><td>Identity between prototype gene and pseudogene at</td></tr>
</table>
</td></tr>
</table>

		the level of DNA
	CDS length	(nt.) CDS length

Rnaspl

Program for predicting exon-exon junction positions in cDNA sequences.

Recognition of exon-exon junctions in cDNA may be very useful for gene sequencing when starting with a sequence of cDNA clone. In a given cDNA sequence we need to select sites for PCR primers that (hopefully) lie in adjacent exons. Prediction is performed by linear discriminant function combining characteristics describing tipical sequences around exon-exon junctions.

Accuracy:

We can not predict exon-exon junction position with very high accuracy, because some important information is being lost during splicing. We predict positions marked by '*', where 75% of potential exon-exon junctions are localized. Additionally, we mark '-' positions where exon-exon junctions atr absent with probability about 90%. We recommend to select primer sequences in continuous '-' regions that do not cross '*' or ' ' positions.

Reference:

Solovyev V.V.,Salamov A.A., Lawrence C.B. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. (Nucl.Acids Res., 1994, 22, 24, 5156-5163).

```
RNASPL output:
First line - name of your sequence
Second line - your sequence
3d line - '*' shows potential exon-exon junction position (Pr > 0.75) '-'
shows position where exon-exon junction absent (Pr > 0.90) 'n' is nonanalyzed
flanking position
For example:
   HSACHG7       690 bp    DNA             PRI       18-DEC-1990
        10        20        30        40        50        60
ATGGCGGCGACGGCGAGTGCCGGGGCCGGCGGATGGACGGGAAGCCCCGTACCTCCCCT
nnnnnnnnnnnnnnnnnnnn--------  ---------*---- ----*----------
        70        80        90       100       110       120
AAGTCCGTCAAGTTCCTGTTTGGGGGCCTGGCCGGGATGGGAGCTACAGTTTTTGTCCAG
----- *----*--------- -- --------*-------  --------------- -
       130       140       150       160       170       180
CCCCTGGACCTGGTGAAGAACCGGATGCAGTTGAGCGGGGAAGGGGCCAAGACTCGAGAG
-----------*-*--- ---- ------ --*----- -----------*------ --
       190       200       210       220       230       240
TACAAAACCAGCTTCCATGCCCTCACCAGTATCCTGAAGGCAGAAGGCCTGAGGGGCATT
------ ---------- ----------------  ------------------------
       250       260       270       280       290       300
TACACTGGGCTGTCGGCTGGCCTGCTGCGTCAGGCCACCTACACCACTACCCGCCTTGGC
----- -- ------------------------------------------------
```

Parameters:

Input	
Sequence	Source file with nucleotide sequences in FASTA format.
Output	
Result file	Name of the output file.

Spl

Prediction of splice sites in Human DNA sequences.

Method description:

Using information about significant triplet frequencies in various functional parts of splice site regions, and preferences of octanucleotides in protein coding and intron regions, a combined linear discriminant recognition function was developed. The splice site prediction scheme gives an accuracy of donor site recognition on the test set 97% (correlation coefficient C=0.62) and 96% for acceptor splice sites (C=0.48). The method is a good alternative to neural network approach (Brunak et al.,Mol.Biol.,1991) that has C=0.61 with 95% accuracy of donor site prediction and C < 40 with 95% accuracy of acceptor site prediction. False positive rate for splice site prediction is relatively high - about one false positive per one true site for 97% accuracy of true sites prediction. More precise splice site positions might be found if you use programs of exons recognition (Fex) and gene structure prediction (Fgenesh).

Spl output:

First line - name of your sequence
Second line - length of your sequence
After that are positions and scores of the predicted sites

For example:

```
HUMALPHA 4556 bp ds-DNA PRI 15-SEP-1
length of sequence - 4556
Number of Donor sites: 11 Threshold: 0.76
1 329 0.76
2 517 0.87
3 728 0.88
4 955 0.98
5 1322 0.81
6 1954 0.85
..............
Number of Acceptor sites: 18 Threshold: 0.65
1 244 0.65
2 379 0.67
3 610 0.89
4 615 0.68
5 838 0.83
6 1146 0.75
...............
```

References:

1. Solovyev V.V.,Salamov A.A., Lawrence C.B. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. (Nucl.Acids Res.,1994,22,24,5156-5163).

2.Solovyev V.V., Salamov A.A. , Lawrence C.B. The prediction of human exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. in: The Second International conference on Intelligent systems for Molecular Biology (eds. Altman R., Brutlag D., Karp R., Latrop R. and Searls D.), AAAI Press, Menlo Park, CA (1994, 354-362)

3. Solovyev V.V., Lawrence C.B. (1993) Identification of Human gene functional regions based on oligonucleotide composition. In Proceedings of First International conference on Intelligent System for Molecular Biology (eds. Hunter L., Searls D., Shalvic J.), Bethesda, 371-379.

Parameters:

Input	
Organism	Select parameter file for specified organizm: **Human** **Drosophila** **C.elegans** **Yeast (S.c.)** **Dicots (Arabidopsis)**
Input file	Browse your source file with nucleotide sequences in FASTA format.
Output	

Output file	Name of the output file.

SplM

Prediction of splice sites in Human DNA sequences.

The program developed by Salamov A and Solovyev V. It locates potential splice site positions based on 5 weight matrices for donor sites and a model including dinucleotide composition and weight matrix for acceptor splice site. Program includes prediction of potential GC -donor sites and non-standard splice sites as AT-AC

Program does not EXCLUDE splice sites close to sites predicted with higher scores or sites on different chains. User could make processing based on the reported scores. It designed to be useful to analyze ALTERNATIVE Splice variants and NON-CANONICAL splice sites. Program has much higher number of overpredicted sites comparing with Spl program.

For some description of this program see:
Solovyev V.V. (2001) Statistical approaches in Eukaryotic gene prediction. In Handbook of Statistical genetics (eds. Balding D. et al.), John Wiley & Sons, Ltd., p. 83-127.

Example of output:

```
Splm: Matrix-based prediction of splice sites in Human sequences
-------------------------------------------------------------------
Parameters: -d 90 -a 90 -dGC 90 -nc 1 (non-st. consensus AT-AC)
 Length of sequence   4500
Number of Donor    sites:     22 Threshold:   90
Number Position Score Chain Type
    1     167      33   -   GT
    2     184      43   -   GC
    3     460      25   -   GT
    4     486      21   -   GC
    5     710      97   +   GT
    6    1077      48   +   GT
    7    1081      18   +   GT
    8    1181      75   -   GT
    9    1920      24   +   GT
   10    2179      36   -   GC
   11    2691      45   +   GT
   12    2745      43   -   GC
   13    2906      18   +   GT
   14    2937      83   +   GT
   15    3006      14   -   GT
   16    3023      90   -   GT
   17    3041      29   -   GT
   18    3107      11   -   GT
   19    3174      46   +   GT
   20    3290      12   -   GT
   21    4156      51   -   GT
   22    4308      22   +   GT
Number of Acceptor sites:     38 Threshold:   90
    1     110      24   -   AG
    2     498      12   +   AG
    3     680      15   +   AG
    4     702      18   -   AG
    5     738      19   +   AG
    6     780      27   -   AG
    7     861      49   +   AG
    8     912      34   -   AG
    9    1033      24   +   AG
   10    1384       8   -   AC
   11    1399      16   +   AG
   12    1780      11   -   AG
   13    1809      14   -   AG
   14    2072      13   +   AG
   15    2120       29   -   AG
   16    2212       61   +   AG
   17    2238       24   -   AG
   18    2258       18   -   AG
   19    2453        8   -   AC
   20    2474       12   -   AG
   21    2508        9   -   AC
   22    2576       94   +   AG
   23    2691        9   -   AC
   24    2755       33   +   AG
   25    2841       41   -   AG
   26    3045        8   +   AC
   27    3108       27   -   AG
   28    3185       14   -   AG
   29    3241       39   +   AG
   30    3267       23   -   AG
   31    3776       25   +   AG
   32    3825       13   -   AG
   33    3885        8   +   AC
   34    4200       12   +   AG
   35    4252       29   +   AG
   36    4290       18   -   AG
   37    4334        9   +   AC
   38    4388       13   +   AG
```

Parameters:

Input	
Input file	Browse your source file with nucleotide sequences in FASTA format.
Output	
Output file	Name of the output file.
Options	
Threshold for donor splice sites	Threshold for donor splice sites (default value 95).
Threshold for acceptor splice sites	Threshold for acceptor splice sites (default value 95).
Threshold for GC donor splice sites	Threshold for GC donor splice sites (default value 95).
Allow search for AT-AC sites	Allow search for AT-AC sites.

PSF-Pre

Finding pseudogenes in a genomic sequence.

Fgenesh++

Pipeline for automatic Eukaryotic genome annotation

Net Blast/Blast

AddProtein

Add known protein sequence from databases that is encoded by a given nucleotide sequence .

Parameters:

Input	
Nucleotide Query Sequence	File with Nucleotide Query Sequence. This should be exactly the same file as for Net-BlastX input.
NetBlastX result file	File with NetBlastX alignments. !NOTE!NetBlastX must be run with output option set to "Pairwise" (Default) style .
Output	
Result	Designates an output file for the search results.
String Length	Specify the nucleotide string length in output file.
Make HTML Output	Make HTML Output.
Show Blast results	Enabling this option specifies if the Blast alignment results will be added to the end of file.
Numeration Style	Numeration style for nucleotides in output file. Three variants are possible: 1. No numeration; 2. To the left of the first nucleotide in a string (Left); 3. Above the each tenth nucleotide in a string (Top).
Options	
Homology threshold	Specifying this parameter, user can discard results with homology percentage lower than set value.
Process first hit only	Enabling this option restricts the output to the first hit only.

AddSNP

Search for known SNPs in a given sequence in NCBI database.

Parameters:

Input	
Nucleotide Query Sequence	File with Nucleotide Query Sequence. This should be exactly the same file as for Net-BlastX input.
DataBase	Select database.
Output	
Result	Designates an output file for the search results.
String Length	Specify the nucleotide string length in output file.
Make HTML Output	Make HTML Output.
Show Blast results	Enabling this option specifies if the Blast alignment results will be added to the end of file.
Numeration Style	Numeration style for nucleotides in output file. Three variants are possible: 1. No numeration;

	2. To the left of the first nucleotide in a string (Left); 3. Above the each tenth nucleotide in a string (Top).
Options	
Query strands	Query strands to search against database.
Process first hit only	Enabling this option restricts the output to the first hit only.

Blast2seq

Blast2seq - BLASTA sequences alignment .
The program aligns sequence (input file) on the base prepared by program FormatDB.
BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.
The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.
The BLAST family of programs allows all combinations of DNA or protein query sequences with searches against DNA or protein databases:

blastp compares an amino acid query sequence against a protein sequence database.

blastn compares a nucleotide query sequence against a nucleotide sequence database.

blastx compares the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database.

tblastn compares a protein query sequence against a nucleotide sequence database dynamically translated in all six reading frames (both strands).

tblastx compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database.

Gaps in Blast

Version 2.0 of BLAST allows the introduction of gaps (deletions and insertions) into alignments. With a gapped alignment tool, homologous domains do not have to be broken into several segments. Also, the scoring of gapped results tends to be more biologically meaningful than ungapped results.
The programs, blastn and blastp, offer fully gapped alignments. blastx and tblastn have 'in-frame' gapped alignments and use sum statistics to link alignments from different frames. tblastx provides only ungapped alignments.

Blast Query Format

The sequence sent to the BLAST server should be in FASTA format, described in http://www.ncbi.nlm.nih.gov/BLAST/fasta.html.
A number of databases are also available. They are described in http://www.ncbi.nlm.nih.gov/BLAST/blast_databases.html.

Reference:

Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Parameters:

Input	
Query sequence	First input file
Target sequence	Second input file
Output	
Result	Designates an output file for the search results.
Options	
Program name	Select search program. 🕐 **Blastp** - compares an amino acid query sequence against a protein sequence database. 🕐 **Blastn** - compares a nucleotide query sequence against a nucleotide sequence database. 🕐 **Blastx** - compares the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database. 🕐 **tBlastn** - compares a protein query sequence against a nucleotide sequence database dynamically translated in all six reading frames (both strands). 🕐 **tBlastx** - compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database. For blastx 1st sequence should be nucleotide, tblastn 2nd sequence nucleotide.
Expectation value	Sets the threshold expectation value for keeping alignments. This is the E from the Karlin-Altschul equation that describes how often an alignment with a given score is expected to occur at random.

BlastN

BlastN compares a nucleotide query sequence against a nucleotide sequence database.
The program aligns sequence (input file) on the base prepared by program FormatDB.
BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.
The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.

Reference:

Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Parameters:

Input	
Blast DB	Identifies the database to search. Database must already be formatted by formatdb.
Nucleotide Query sequence(s)	If the input file contains multiple sequences, BLAST will be run on each sequence in order, and the resulting output will contain concatenated BLAST reports.
Believe the query defline.	Believe the query definition line.
Output	
Result	Designates an output file for the search results.
Format	Pairwise (Default) Query-anchored, showing identities Query-anchored, no identities

	Flat query-anchored, showing identities Flat query-anchored, no identities Query-anchored, no identities and blunt ends Flat query-anchored, no identities and blunt ends XML Blast output Tabular Tabular with comment lines ASN, text ASN, binary
Show GI's in deflines	Shows GenInfo Identifier (GI) numbers in definition lines. A GI is a unique numeric identifier assigned for a sequence in GenBank. A GI corresponds to an accession version pair.
Produce HTML output	Produces HTML output with [anchor] links from the summary at the top of the report to the alignments farther below. This option should be used only with the standard report format ("Pairwise (Default)").
Number of Alignments to output	Truncates the report to set number of alignments. There is no warning when you exceed this limit, so it's generally a good idea to set this value very high unless you're interested only in the top hits.
SeqAlign file (Optional)	SeqAlign output file
Options	
MegaBlast search	Sets the blastn program to the megablast mode, which is optimized to find near identities very quickly.
Expectation value	Sets the threshold expectation value for keeping alignments. This is the E from the Karlin-Altschul equation that describes how often an alignment with a given score is expected to occur at random.
Filter query sequence	Filters the query sequence for low-complexity subsequences. The default setting is ON. Complexity filtering is generally a good idea, but it may break long HSPs into several smaller HSPs due to low-complexity segments. This can cause some alignments to fall below the significance threshold and be lost. To prevent this, either turn off filtering (not recommended) or use soft masking, in which the filter is used only in the word seeding phase, but not the extension phase. DUST with blastn, SEG with others.
Perform gapped alignment	Performs gapped alignment. Setting this to OFF invokes the older, ungapped style of alignment. You can't perform gapped alignments with tblastx, regardless of this setting.
Open Gap Cost	Initial penalty for opening a gap of length 0. -1 invokes the default behavior, and setting the parameter to zero is impossible, unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap costs for programs other than blastn depend on the scoring matrix.
Extend Gap Cost	The penalty for each gap character. Note that value -1 is synonymous with the default behavior for the "Open Gap Cost" parameter and, it's impossible to set value to zero unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap cost, for programs other than blastn, depends on the scoring matrix.
Gapped Alignment X dropoff value	X dropoff value for gapped alignment (in bits); Zero invokes default behavior; blastn 30, megablast 20, tblastx 0, all others 15.

Nucleotide Mismatch Penalty	Sets the penalty for a nucleotide mismatch. Also see "Nucleotide Match Reward". The choice of [integer] for "Nucleotide Mismatch Penalty" and "Nucleotide Match Reward" are very important because they determine your target frequencies. The default values 1 for "Nucleotide Match Reward" and -3 for "Nucleotide Mismatch Penalty" are most effective for aligning sequences that are 99 percent identical.
Nucleotide Match Reward	Sets the score of a nucleotide match. See also the "Nucleotide Mismatch Penalty" parameter.
Number of DB Seqs to show descriptions	Sets the number of database sequences for which to show the one-line summary descriptions at the top of a BLAST report. You won't be warned if you exceed a value. Also see the "Number of Alignments to output" parameter.
Extending Hits Threshold	Neighborhood word threshold score. Only those words scoring equal to or greater than [value] will seed alignments. Zero is default; blastp 11, blastn 0, blastx 12, tblastn 13, tblastx 13, megablast 0.
Word size	Sets the word size for the initial word search. The minimum word size for blastn is 7.
DataBase Effective Length	Effective length of the database. Use zero for the real size (Default).
Best Hits Number	The number of best hits from a region to keep. This option is useful when you want to limit the number of alignments that might pile up in one section of the query. This is most useful if the settings of "Number of Alignments to output" or "Number of DB Seqs to show descriptions" are low, and the abundant alignments push lower scoring alignments off the end of the report. Off by default, if used a value of 100 is recommended.
Two-hit or Single-hit Algorithm	Specifies the two-hit or single-hit algorithm. The two-hit option requires two word hits on the same diagonal to extend from either one. When set to two-hit mode, the "Multiple Hits Window Size" parameter specifies how close the two hits have to be to trigger extension.
Query strands	Chooses which strand of DNA-based queries is searched. **Top Strand** **Bottom Strand** **Both Strands**
Location on query sequence	The location on query sequence. This lets you limit the search to a subsequence of the query sequence. For example, to search just the letters from 21 to 50, set the parameter to following: "21,50" The alignments won't extend outside the specified region. In older versions of BLAST, this parameter set the size of the region under control of the "Best Hits Number" parameter.
Search Space Effective Length	Effective length of the search space. Use zero for the real size (Default).
Lower Case Filtering	Use lower case filtering of FASTA sequence.
Ungapped Extension X	X dropoff value for ungapped extensions in bits; Zero invokes default behavior; blastn 20, megablast 10, all others 7.

dropoff value	
Final Gapped Alignment X dropoff value	X dropoff value for final gapped alignment in bits; Sets the X3 dropoff value (in bits) for extensions but is bounded by the value for X2. Zero invokes default behavior; blastn/megablast 50, tblastx 0, all others 25.
Multiple Hits Window Size	Sets the multiple-hit window size [integer]. When BLAST is set to two-hit mode, this option requires two word hits on the same diagonal to be within [value] letters of each other in order to extend from either one. The larger the [value], the more sensitive BLAST will be. Setting [value] to 0 sets the default behavior of 40, except for blastn, whose default is single word hit. To specify one-hit behavior, set 1. Blastn/megablast 0 (Default), all others 40.
Concatenated Queries Number	Sets the number of queries to concatenate in a single search [integer]. Concatenating queries accelerates the search because the database is scanned just one time. The specified value must be the number of sequences in the query file. if it's less, only the first set of [value] sequences is used. Also, the output is very different than you would expect. All the query names are listed, and then all the one-line summaries are given, followed by the alignments, and finally, one footer is produced for the whole report. Given this format, it's very difficult to discern which alignments belong to which query. This option should not be used in its current implementation.
Number of processors	Sets the number of processors to use. If you have multiple queries, you will get better throughput by executing multiple BLAST searches. For insensitive searches such as default BLASTN, setting -a to a higher value may not appreciably improve speed if disk I/O is the bottleneck.
Old Engine Use	Force use of old engine.

BlastP

BlastP compares an amino acid query sequence against a protein sequence database.
The program aligns sequence (input file) on the base prepared by program FormatDB.
BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.
The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.

Reference:

Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Parameters:

Input	
Blast DB	Identifies the database to search. Database must already be formatted by formatdb.
Protein Query sequence(s)	If the input file contains multiple sequences, BLAST will be run on each sequence in order, and the resulting output will contain concatenated BLAST reports.

Believe the query defline.	Believe the query definition line.
Output	
Result	Designates an output file for the search results.
Format	Pairwise (Default) Query-anchored, showing identities Query-anchored, no identities Flat query-anchored, showing identities Flat query-anchored, no identities Query-anchored, no identities and blunt ends Flat query-anchored, no identities and blunt ends XML Blast output Tabular Tabular with comment lines ASN, text ASN, binary
Show GI's in deflines	Shows GenInfo Identifier (GI) numbers in definition lines. A GI is a unique numeric identifier assigned for a sequence in GenBank. A GI corresponds to an accession version pair.
Produce HTML output	Produces HTML output with [anchor] links from the summary at the top of the report to the alignments farther below. This option should be used only with the standard report format ("Pairwise (Default)").
Number of Alignments to output	Truncates the report to set number of alignments. There is no warning when you exceed this limit, so it's generally a good idea to set this value very high unless you're interested only in the top hits.
SeqAlign file (Optional)	SeqAlign output file
Options	
Expectation value	Sets the threshold expectation value for keeping alignments. This is the E from the Karlin-Altschul equation that describes how often an alignment with a given score is expected to occur at random.
Filter query sequence	Filters the query sequence for low-complexity subsequences. The default setting is ON. Complexity filtering is generally a good idea, but it may break long HSPs into several smaller HSPs due to low-complexity segments. This can cause some alignments to fall below the significance threshold and be lost. To prevent this, either turn off filtering (not recommended) or use soft masking, in which the filter is used only in the word seeding phase, but not the extension phase. DUST with blastn, SEG with others.
Perform gapped alignment	Performs gapped alignment. Setting this to OFF invokes the older, ungapped style of alignment. You can't perform gapped alignments with tblastx, regardless of this setting.
Open Gap Cost	Initial penalty for opening a gap of length 0. -1 invokes the default behavior, and setting the parameter to zero is impossible, unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap costs for programs other than blastn depend on the scoring matrix.
Extend Gap Cost	The penalty for each gap character. Note that value -1 is synonymous with the default behavior for the "Open Gap Cost" parameter and, it's impossible to set

	value to zero unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap cost, for programs other than blastn, depends on the scoring matrix.
Gapped Alignment X dropoff value	X dropoff value for gapped alignment (in bits); Zero invokes default behavior; blastn 30, megablast 20, tblastx 0, all others 15.
Number of DB Seqs to show descriptions	Sets the number of database sequences for which to show the one-line summary descriptions at the top of a BLAST report. You won't be warned if you exceed a value. Also see the "Number of Alignments to output" parameter.
Extending Hits Threshold	Neighborhood word threshold score. Only those words scoring equal to or greater than [value] will seed alignments. Zero is default; blastp 11, blastn 0, blastx 12, tblastn 13, tblastx 13, megablast 0.
Matrix	Designates a protein similarity matrix. This is used in all BLAST programs except blastn. Matrices are sought in the following order: in the local directory, in the location specified in the .ncbirc file, in a local data directory, and finally, in the BLASTMAT environment variable (only on Unix systems). Other matrices included in the standard distribution include BLOSUM45, BLOSUM80, PAM30, and PAM70. You can use custom matrix files, but it requires modifying the source code and defining the new matrix with all of its associated statistics for different affine gap combinations and recompiling the binary. Using these custom files isn't recommended because it requires the arduous task of calculating gapped values for lambda and maintaining a derivative branch of the source code.
Word size	Sets the word size for the initial word search. Word sizes for blastp, blastx, tblastn, and tblastx are 2 or 3.
DataBase Effective Length	Effective length of the database. Use zero for the real size (Default).
Best Hits Number	The number of best hits from a region to keep. This option is useful when you want to limit the number of alignments that might pile up in one section of the query. This is most useful if the settings of "Number of Alignments to output" or "Number of DB Seqs to show descriptions" are low, and the abundant alignments push lower scoring alignments off the end of the report. Off by default, if used a value of 100 is recommended.
Two-hit or Single-hit Algorithm	Specifies the two-hit or single-hit algorithm. The two-hit option requires two word hits on the same diagonal to extend from either one. When set to two-hit mode, the "Multiple Hits Window Size" parameter specifies how close the two hits have to be to trigger extension.
Location on query sequence	The location on query sequence. This lets you limit the search to a subsequence of the query sequence. For example, to search just the letters from 21 to 50, set the parameter to following: "21,50". The alignments won't extend outside the specified region. In older versions of BLAST, this parameter set the size of the region under control of the "Best Hits Number" parameter.
Search Space	Effective length of the search space. Use zero for the real size (Default).

Effective Length	
Lower Case Filtering	Use lower case filtering of FASTA sequence.
Ungapped Extension X dropoff value	X dropoff value for ungapped extensions in bits; Zero invokes default behavior; blastn 20, megablast 10, all others 7.
Final Gapped Alignment X dropoff value	X dropoff value for final gapped alignment in bits; Sets the X3 dropoff value (in bits) for extensions but is bounded by the value for X2. Zero invokes default behavior; blastn/megablast 50, tblastx 0, all others 25.
Multiple Hits Window Size	Sets the multiple-hit window size [integer]. When BLAST is set to two-hit mode, this option requires two word hits on the same diagonal to be within [value] letters of each other in order to extend from either one. The larger the [value], the more sensitive BLAST will be. Setting [value] to 0 sets the default behavior of 40, except for blastn, whose default is single word hit. To specify one-hit behavior, set 1. Blastn/megablast 0 (Default), all others 40.
Number of processors	Sets the number of processors to use. If you have multiple queries, you will get better throughput by executing multiple BLAST searches. For insensitive searches such as default BLASTN, setting -a to a higher value may not appreciably improve speed if disk I/O is the bottleneck.
Old Engine Use	Force use of old engine.

BlastX

Compares a nucleotide query sequence against a nucleotide sequence database.
BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.
The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.

Reference:

Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Parameters:

Input	
Blast DB	Identifies the database to search. Database must already be formatted by formatdb.
Nucleotide Query sequence(s)	If the input file contains multiple sequences, BLAST will be run on each sequence in order, and the resulting output will contain concatenated BLAST reports.
Believe the query defiline.	Believe the query definition line.
Output	
Result	Designates an output file for the search results.
Format	Pairwise (Default) Query-anchored, showing identities

	Query-anchored, no identities Flat query-anchored, showing identities Flat query-anchored, no identities Query-anchored, no identities and blunt ends Flat query-anchored, no identities and blunt ends XML Blast output Tabular Tabular with comment lines ASN, text ASN, binary
Show GI's in deflines	Shows GenInfo Identifier (GI) numbers in definition lines. A GI is a unique numeric identifier assigned for a sequence in GenBank. A GI corresponds to an accession version pair.
Produce HTML output	Produces HTML output with [anchor] links from the summary at the top of the report to the alignments farther below. This option should be used only with the standard report format ("Pairwise (Default)").
Number of Alignments to output	Truncates the report to set number of alignments. There is no warning when you exceed this limit, so it's generally a good idea to set this value very high unless you're interested only in the top hits.
SeqAlign file (Optional)	SeqAlign output file
Options	
Expectation value	Sets the threshold expectation value for keeping alignments. This is the E from the Karlin-Altschul equation that describes how often an alignment with a given score is expected to occur at random.
Filter query sequence	Filters the query sequence for low-complexity subsequences. The default setting is ON. Complexity filtering is generally a good idea, but it may break long HSPs into several smaller HSPs due to low-complexity segments. This can cause some alignments to fall below the significance threshold and be lost. To prevent this, either turn off filtering (not recommended) or use soft masking, in which the filter is used only in the word seeding phase, but not the extension phase. DUST with blastn, SEG with others.
Perform gapped alignment	Performs gapped alignment. Setting this to OFF invokes the older, ungapped style of alignment. You can't perform gapped alignments with tblastx, regardless of this setting.
Open Gap Cost	Initial penalty for opening a gap of length 0. -1 invokes the default behavior, and setting the parameter to zero is impossible, unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap costs for programs other than blastn depend on the scoring matrix.
Extend Gap Cost	The penalty for each gap character. Note that value -1 is synonymous with the default behavior for the "Open Gap Cost" parameter and, it's impossible to set value to zero unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap cost, for programs other than blastn, depends on the scoring matrix.
Gapped Alignment X dropoff value	X dropoff value for gapped alignment (in bits); Zero invokes default behavior; blastn 30, megablast 20, tblastx 0, all others 15.
Number of DB	Sets the number of database sequences for which to show the one-line summary

Seqs to show descriptions	descriptions at the top of a BLAST report. You won't be warned if you exceed a value. Also see the "Number of Alignments to output" parameter.
Extending Hits Threshold	Neighborhood word threshold score. Only those words scoring equal to or greater than [value] will seed alignments. Zero is default; blastp 11, blastn 0, blastx 12, tblastn 13, tblastx 13, megablast 0.
Translation table	Select translation table.
Matrix	Designates a protein similarity matrix. This is used in all BLAST programs except blastn. Matrices are sought in the following order: in the local directory, in the location specified in the .ncbirc file, in a local data directory, and finally, in the BLASTMAT environment variable (only on Unix systems). Other matrices included in the standard distribution include BLOSUM45, BLOSUM80, PAM30, and PAM70. You can use custom matrix files, but it requires modifying the source code and defining the new matrix with all of its associated statistics for different affine gap combinations and recompiling the binary. Using these custom files isn't recommended because it requires the arduous task of calculating gapped values for lambda and maintaining a derivative branch of the source code.
Word size	Sets the word size for the initial word search. Word sizes for blastp, blastx, tblastn, and tblastx are 2 or 3.
DataBase Effective Length	Effective length of the database. Use zero for the real size (Default).
Best Hits Number	The number of best hits from a region to keep. This option is useful when you want to limit the number of alignments that might pile up in one section of the query. This is most useful if the settings of "Number of Alignments to output" or "Number of DB Seqs to show descriptions" are low, and the abundant alignments push lower scoring alignments off the end of the report. Off by default, if used a value of 100 is recommended.
Two-hit or Single-hit Algorithm	Specifies the two-hit or single-hit algorithm. The two-hit option requires two word hits on the same diagonal to extend from either one. When set to two-hit mode, the "Multiple Hits Window Size" parameter specifies how close the two hits have to be to trigger extension.
Query strands	Chooses which strand of DNA-based queries is searched. **Top Strand** **Bottom Strand** **Both Strands**
Location on query sequence	The location on query sequence. This lets you limit the search to a subsequence of the query sequence. For example, to search just the letters from 21 to 50, set the parameter to following: "21,50". The alignments won't extend outside the specified region. In older versions of BLAST, this parameter set the size of the region under control of the "Best Hits Number" parameter.
Search Space Effective Length	Effective length of the search space. Use zero for the real size (Default).
Lower Case	Use lower case filtering of FASTA sequence.

Filtering	
Ungapped Extension X dropoff value	X dropoff value for ungapped extensions in bits; Zero invokes default behavior; blastn 20, megablast 10, all others 7.
Final Gapped Alignment X dropoff value	X dropoff value for final gapped alignment in bits; Sets the X3 dropoff value (in bits) for extensions but is bounded by the value for X2. Zero invokes default behavior; blastn/megablast 50, tblastx 0, all others 25.
Multiple Hits Window Size	Sets the multiple-hit window size [integer]. When BLAST is set to two-hit mode, this option requires two word hits on the same diagonal to be within [value] letters of each other in order to extend from either one. The larger the [value], the more sensitive BLAST will be. Setting [value] to 0 sets the default behavior of 40, except for blastn, whose default is single word hit. To specify one-hit behavior, set 1. Blastn/megablast 0 (Default), all others 40.

Frame shift penalty	Sets the frame shift penalty for the Out Of Frame (OOF) algorithm of blastx.
	When the parameter is set, it invokes the OOF mode of BLAST, which lets alignments proceed across reading frames. The expect values calculated from OOF blastx are only approximate, and BLAST issues the following warning when OOF is invoked: [NULL_Caption] WARNING: test500: Out-of-frame option selected, Expect values are only approximate and calculated not assuming out-of-frame alignments
	The out-of-frame alignments are signified by slashes that indicate the +1(/),+2(//), -1(\), and -2(\\) frameshifts. The following is a sample OOF alignment:
	`Query: 23  PLIRNSL/YCINC\\A//QSIIRAHVKGPYLTRWVVNC/E\TCSKGYAKTPGASTDLLLL 160`
	`           PLIRNSL YCINC     QSIIRAHVKGPYLTRWVVNC   TCSKGYAKTPGASTDLLLL`
	`Sbjct: 1   PLIRNSL YCINC  X  QSIIRAHVKGPYLTRWVVNC X TCSKGYAKTPGASTDLLLL 53`
	`Query: 161 YKTRNSLTSASSLSPVRSQRMI/N\SFPRFQGHLVVSG/S\SAHNR/FS\FNRDSPRGSG 322`
	`           YKTRNSLTSASSLSPVRSQRMI   SFPRFQGHLVVSG   SAHNR F  FNRDSPRGSG`
	`Sbjct: 54  YKTRNSLTSASSLSPVRSQRMI X SFPRFQGHLVVSG X SAHNR FX FNRDSPRGSG 107`
	`Query: 323 SYCSREPMGQIKIRRTHTDDKLFR/ND\SRHTRAGDGLNI//TLA\\RDPSFLSRVYNAN 484`
	`           SYCSREPMGQIKIRRTHTDDKLFR    SRHTRAGDGLNI   L   RDPSFLSRVYNAN`
	`Sbjct: 108 SYCSREPMGQIKIRRTHTDDKLFR XX SRHTRAGDGLNI  XLX  RDPSFLSRVYNAN 161`
	`Query: 485 SYLHI 499`
	`           SYLHI`
	`Sbjct: 162 SYLHI 166`

Number of processors	Sets the number of processors to use. If you have multiple queries, you will get better throughput by executing multiple BLAST searches. For insensitive searches such as default BLASTN, setting -a to a higher value may not appreciably improve speed if disk I/O is the bottleneck.
Old Engine Use	Force use of old engine.

tBlastN

tBlastN compares a nucleotide query sequence against a nucleotide sequence database.

BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.

The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.

Reference:

Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Parameters:

Input	
Blast DB	Identifies the database to search. Database must already be formatted by formatdb.
Protein Query sequence(s)	If the input file contains multiple sequences, BLAST will be run on each sequence in order, and the resulting output will contain concatenated BLAST reports.
Believe the query defline.	Believe the query definition line.
Output	
Result	Designates an output file for the search results.
Format	Pairwise (Default) Query-anchored, showing identities Query-anchored, no identities Flat query-anchored, showing identities Flat query-anchored, no identities Query-anchored, no identities and blunt ends Flat query-anchored, no identities and blunt ends XML Blast output Tabular Tabular with comment lines ASN, text ASN, binary
Show GI's in deflines	Shows GenInfo Identifier (GI) numbers in definition lines. A GI is a unique numeric identifier assigned for a sequence in GenBank. A GI corresponds to an accession version pair.
Produce HTML output	Produces HTML output with [anchor] links from the summary at the top of the report to the alignments farther below. This option should be used only with the standard report format ("Pairwise (Default)").
Number of Alignments to output	Truncates the report to set number of alignments. There is no warning when you exceed this limit, so it's generally a good idea to set this value very high unless you're interested only in the top hits.
SeqAlign file (Optional)	SeqAlign output file
Options	
Expectation value	Sets the threshold expectation value for keeping alignments. This is the E from the Karlin-Altschul equation that describes how often an alignment with a given score is expected to occur at random.
Filter query sequence	Filters the query sequence for low-complexity subsequences. The default setting is ON. Complexity filtering is generally a good idea, but it may break long HSPs into several smaller HSPs due to low-complexity segments. This can cause some alignments to fall below the significance threshold and be lost. To prevent this, either turn off filtering (not recommended) or use soft masking, in which the filter is used only in the word seeding phase, but not the extension phase.

	DUST with blastn, SEG with others.
Perform gapped alignment	Performs gapped alignment. Setting this to OFF invokes the older, ungapped style of alignment. You can't perform gapped alignments with tblastx, regardless of this setting.
Open Gap Cost	Initial penalty for opening a gap of length 0. -1 invokes the default behavior, and setting the parameter to zero is impossible, unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap costs for programs other than blastn depend on the scoring matrix.
Extend Gap Cost	The penalty for each gap character. Note that value -1 is synonymous with the default behavior for the "Open Gap Cost" parameter and, it's impossible to set value to zero unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap cost, for programs other than blastn, depends on the scoring matrix.
Smith-Waterman alignments	Compute locally optimal Smith-Waterman alignments. This option is only available for gapped tblastn.
Gapped Alignment X dropoff value	X dropoff value for gapped alignment (in bits); Zero invokes default behavior; blastn 30, megablast 20, tblastx 0, all others 15.
Number of DB Seqs to show descriptions	Sets the number of database sequences for which to show the one-line summary descriptions at the top of a BLAST report. You won't be warned if you exceed a value. Also see the "Number of Alignments to output" parameter.
Extending Hits Threshold	Neighborhood word threshold score. Only those words scoring equal to or greater than [value] will seed alignments. Zero is default; blastp 11, blastn 0, blastx 12, tblastn 13, tblastx 13, megablast 0.
DB Genetic code	The genetic code to use for translation of the database nucleotide sequence. See http://www.ncbi.nlm.nih.gov/htbin-post/Taxonomy for updates
Matrix	Designates a protein similarity matrix. This is used in all BLAST programs except blastn. Matrices are sought in the following order: in the local directory, in the location specified in the .ncbirc file, in a local data directory, and finally, in the BLASTMAT environment variable (only on Unix systems). Other matrices included in the standard distribution include BLOSUM45, BLOSUM80, PAM30, and PAM70. You can use custom matrix files, but it requires modifying the source code and defining the new matrix with all of its associated statistics for different affine gap combinations and recompiling the binary. Using these custom files isn't recommended because it requires the arduous task of calculating gapped values for lambda and maintaining a derivative branch of the source code.
Word size	Sets the word size for the initial word search. Word sizes for blastp, blastx, tblastn, and tblastx are 2 or 3.
DataBase Effective Length	Effective length of the database. Use zero for the real size (Default).
Best Hits Number	The number of best hits from a region to keep. This option is useful when you want to limit the number of alignments that might pile up in one section of the query. This is most useful if the settings of "Number of Alignments to output" or "Number of DB Seqs to show descriptions" are low, and the abundant alignments push lower scoring alignments off the end of the report. Off by default, if used a value of 100 is recommended.

Two-hit or Single-hit Algorithm	Specifies the two-hit or single-hit algorithm. The two-hit option requires two word hits on the same diagonal to extend from either one. When set to two-hit mode, the "Multiple Hits Window Size" parameter specifies how close the two hits have to be to trigger extension.
Location on query sequence	The location on query sequence. This lets you limit the search to a subsequence of the query sequence. For example, to search just the letters from 21 to 50, set the parameter to following: "21,50". The alignments won't extend outside the specified region. In older versions of BLAST, this parameter set the size of the region under control of the "Best Hits Number" parameter.
Search Space Effective Length	Effective length of the search space. Use zero for the real size (Default).
Lower Case Filtering	Use lower case filtering of FASTA sequence.
Ungapped Extension X dropoff value	X dropoff value for ungapped extensions in bits; Zero invokes default behavior; blastn 20, megablast 10, all others 7.
Final Gapped Alignment X dropoff value	X dropoff value for final gapped alignment in bits; Sets the X3 dropoff value (in bits) for extensions but is bounded by the value for X2. Zero invokes default behavior; blastn/megablast 50, tblastx 0, all others 25.
Multiple Hits Window Size	Sets the multiple-hit window size [integer]. When BLAST is set to two-hit mode, this option requires two word hits on the same diagonal to be within [value] letters of each other in order to extend from either one. The larger the [value], the more sensitive BLAST will be. Setting [value] to 0 sets the default behavior of 40, except for blastn, whose default is single word hit. To specify one-hit behavior, set 1. Blastn/megablast 0 (Default), all others 40.
Largest Intron Length	Length of the largest intron allowed in tblastn for linking HSPs. A default of 0 means that linking is turned off.
Concatenated Queries Number	Sets the number of queries to concatenate in a single search [integer]. Concatenating queries accelerates the search because the database is scanned just one time. The specified value must be the number of sequences in the query file. if it's less, only the first set of [value] sequences is used. Also, the output is very different than you would expect. All the query names are listed, and then all the one-line summaries are given, followed by the alignments, and finally, one footer is produced for the whole report. Given this format, it's very difficult to discern which alignments belong to which query. This option should not be used in its current implementation.
Composition-based statistics	Use composition-based statistics for tblastn. For programs other than tblastn, must be absent (Default). Possible choices: 1. Composition-based statistics as in NAR 29:2994-3005, 2001. 2. Composition-based score adjustment as in Bioinformatics 21:902-911, 2005, conditioned on sequence properties. 3. Composition-based score adjustment as in Bioinformatics 21:902-911, 2005, unconditionally.

Number of processors	Sets the number of processors to use. If you have multiple queries, you will get better throughput by executing multiple BLAST searches. For insensitive searches such as default BLASTN, setting -a to a higher value may not appreciably improve speed if disk I/O is the bottleneck.
Old Engine Use	Force use of old engine.

tBlastX

tBlastX compares a nucleotide query sequence against a nucleotide sequence database.
BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.
The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.

Reference:

Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Parameters:

Input	
Blast DB	Identifies the database to search. Database must already be formatted by formatdb.
Nucleotide Query sequence(s)	If the input file contains multiple sequences, BLAST will be run on each sequence in order, and the resulting output will contain concatenated BLAST reports.
Believe the query defline.	Believe the query definition line.
Output	
Result	Designates an output file for the search results.
Format	Pairwise (Default) Query-anchored, showing identities Query-anchored, no identities Flat query-anchored, showing identities Flat query-anchored, no identities Query-anchored, no identities and blunt ends Flat query-anchored, no identities and blunt ends XML Blast output Tabular Tabular with comment lines ASN, text ASN, binary
Show GI's in deflines	Shows GenInfo Identifier (GI) numbers in definition lines. A GI is a unique numeric identifier assigned for a sequence in GenBank. A GI corresponds to an accession version pair.
Produce HTML output	Produces HTML output with [anchor] links from the summary at the top of the report to the alignments farther below. This option should be used only with the standard report format ("Pairwise (Default)").
Number of Alignments to	Truncates the report to set number of alignments. There is no warning when you exceed this limit, so it's generally a good idea to

output	set this value very high unless you're interested only in the top hits.
SeqAlign file (Optional)	SeqAlign output file
Options	
Expectation value	Sets the threshold expectation value for keeping alignments. This is the E from the Karlin-Altschul equation that describes how often an alignment with a given score is expected to occur at random.
Filter query sequence	Filters the query sequence for low-complexity subsequences. The default setting is ON. Complexity filtering is generally a good idea, but it may break long HSPs into several smaller HSPs due to low-complexity segments. This can cause some alignments to fall below the significance threshold and be lost. To prevent this, either turn off filtering (not recommended) or use soft masking, in which the filter is used only in the word seeding phase, but not the extension phase. DUST with blastn, SEG with others.
Gapped Alignment X dropoff value	X dropoff value for gapped alignment (in bits); Zero invokes default behavior; blastn 30, megablast 20, tblastx 0, all others 15.
Number of DB Seqs to show descriptions	Sets the number of database sequences for which to show the one-line summary descriptions at the top of a BLAST report. You won't be warned if you exceed a value. Also see the "Number of Alignments to output" parameter.
Extending Hits Threshold	Neighborhood word threshold score. Only those words scoring equal to or greater than [value] will seed alignments. Zero is default; blastp 11, blastn 0, blastx 12, tblastn 13, tblastx 13, megablast 0.
Translation table	Select translation table.
DB Genetic code	The genetic code to use for translation of the database nucleotide sequence. See http://www.ncbi.nlm.nih.gov/htbin-post/Taxonomy for updates
Matrix	Designates a protein similarity matrix. This is used in all BLAST programs except blastn. Matrices are sought in the following order: in the local directory, in the location specified in the .ncbirc file, in a local data directory, and finally, in the BLASTMAT environment variable (only on Unix systems). Other matrices included in the standard distribution include BLOSUM45, BLOSUM80, PAM30, and PAM70. You can use custom matrix files, but it requires modifying the source code and defining the new matrix with all of its associated statistics for different affine gap combinations and recompiling the binary. Using these custom files isn't recommended because it requires the arduous task of calculating gapped values for lambda and maintaining a derivative branch of the source code.
Word size	Sets the word size for the initial word search. Word sizes for blastp, blastx, tblastn, and tblastx are 2 or 3.
DataBase Effective Length	Effective length of the database. Use zero for the real size (Default).
Best Hits Number	The number of best hits from a region to keep. This option is useful when you want to limit the number of alignments that might pile up in one section of the query. This is most useful if the settings of

	"Number of Alignments to output" or "Number of DB Seqs to show descriptions" are low, and the abundant alignments push lower scoring alignments off the end of the report. Off by default, if used a value of 100 is recommended.
Two-hit or Single-hit Algorithm	Specifies the two-hit or single-hit algorithm. The two-hit option requires two word hits on the same diagonal to extend from either one. When set to two-hit mode, the "Multiple Hits Window Size" parameter specifies how close the two hits have to be to trigger extension.
Query strands	Chooses which strand of DNA-based queries is searched. **Top Strand** **Bottom Strand** **Both Strands**
Location on query sequence	The location on query sequence. This lets you limit the search to a subsequence of the query sequence. For example, to search just the letters from 21 to 50, set the parameter to following: "21,50". The alignments won't extend outside the specified region. In older versions of BLAST, this parameter set the size of the region under control of the "Best Hits Number" parameter.
Search Space Effective Length	Effective length of the search space. Use zero for the real size (Default).
Lower Case Filtering	Use lower case filtering of FASTA sequence.
Ungapped Extension X dropoff value	X dropoff value for ungapped extensions in bits; Zero invokes default behavior; blastn 20, megablast 10, all others 7.
Final Gapped Alignment X dropoff value	X dropoff value for final gapped alignment in bits; Sets the X3 dropoff value (in bits) for extensions but is bounded by the value for X2. Zero invokes default behavior; blastn/megablast 50, tblastx 0, all others 25.
Multiple Hits Window Size	Sets the multiple-hit window size [integer]. When BLAST is set to two-hit mode, this option requires two word hits on the same diagonal to be within [value] letters of each other in order to extend from either one. The larger the [value], the more sensitive BLAST will be. Setting [value] to 0 sets the default behavior of 40, except for blastn, whose default is single word hit. To specify one-hit behavior, set 1. Blastn/megablast 0 (Default), all others 40.
Number of processors	Sets the number of processors to use. If you have multiple queries, you will get better throughput by executing multiple BLAST searches. For insensitive searches such as default BLASTN, setting -a to a higher value may not appreciably improve speed if disk I/O is the bottleneck.
Old Engine Use	Force use of old engine.

FormatDB

Prepare bases for BLAST search.

BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.

The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.

FormatDB, should be used to format the FASTA databases for both protein and DNA databases for BLAST 2.0. This must be done before blastall or blastpgp can be run locally. The format of the databases has been changed substantially from the BLAST 1.4 release. A major improvement in this format over the old one is that ambiguity information for DNA sequences is now retrieved from the files produced by FormatDB, rather than from the original FASTA file. The original FASTA file is no longer needed for the BLAST runs. FormatDB may be obtained with the other BLAST binaries from the executables directory (see above). The input for FormatDB may be either ASN.1 or FASTA. Use of ASN.1 is advantageous for those sites that might also wish to format the ASN.1 in different ways, such as a GenBank report. Usage of FormatDB may be obtained by executing FormatDB and a dash.

References

Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Karlin, Samuel and Stephen F. Altschul (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. Proc. Natl. Acad. Sci. USA 87:2264-68.

Karlin, Samuel and Stephen F. Altschul (1993). Applications and statistics for multiple high-scoring segments in molecu- lar sequences. Proc. Natl. Acad. Sci. USA 90:5873-7.

Parameters:

	Input
Sequences set	Sequences set
Format	Input file format: **Protein** **Nucleotide**
	Output
Result	Name of the output file.
	Output
Parse option	Parse option: **Parse SeqId** - Parse SeqId and create indexes. **Do not parse SeqId** - Do not parse SeqId. Do not create indexes.

NetBlastN

BLASTA Nucleotide search program (net search)
Variant of the BlastN program intended for work with distant databases.

BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.

The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.

Reference:

Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Please, pay attention to following recommendations NCBI (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/firewall.html):

When first downloaded, your NCBI application runs in stand-alone mode, without access to the network. However, your program can also be configured to exchange information with the NCBI (GenBank) over the Internet. The network-aware mode of your application is identical to the stand-alone mode, but it contains some additional useful options.

Your application can only function in its network-aware mode if the computer on which it resides has a direct Internet connection. Electronic mail access to the Internet is insufficient. In general, if you can install and use a WWW-browser on your system, you should be able to install and use the network. Check with your system administrator or Internet provider if you are uncertain as to whether you have direct Internet connectivity.

To launch the configuration form, select Net Configure under the Misc menu in Sequin or Network Entrez, or the Options menu in Cn3D. If you are using blastcl3, you must run Sequin, Network Entrez, or Cn3D first to configure blastcl3. This is necessary because blastcl3 has no graphical user interface.

If you are not behind a firewall, set the **Connection** control to **Normal**. If you also have a Domain Name Server (DNS) available, you can now simply press **Accept**.

If DNS is not available, uncheck the **Domain Name Server** button. If you are behind a firewall, set the **Connection** control to **Firewall**. The **Proxy** box then becomes active. If you also use a proxy server, type in its address. (If you have DNS, it will be of the form `www.myproxy.myuniversity.edu`. If you do not have DNS, you should use the numerical IP address of the form `127.45.23.6`.) Once you type something in the **Proxy** box, the **Port** box and **Transparent Proxy** button become active and can be filled in or changed as appropriate. (By default the **Transparent Proxy** button is off, indicating a CERN-like proxy.) Ask your network administrator for advice on the proper settings to use.

If you are in the United States, the default **Timeout** of 30 seconds should suffice. From foreign countries with poor Internet connection to the U.S., you can select up to 5 minutes as the timeout.

Finally, you will need to quit and restart your application in order for the network-aware settings to take effect.

If you are behind a firewall, it must be configured correctly to access NCBI services. Your network administrators may have done this already. If not, please have them read the section below.

The following section is intended for network administrators:

Using NCBI services from behind a security firewall requires opening ports in your firewall. The ports to open are:

```
Firewall Port        IP Address
--------------------------------
 5860..5870         130.14.29.112
    5845            130.14.22.12  (cannot be accessed from outside NCBI!)
```

If your firewall is not transparent, the firewall port number should be mapped to the same port number on the external host.

Port 5860 is usually not accessible by the public but reserved for NCBI internal purposes only. However, we recommend that it is kept open just as all other ports in the range in case the public access will be eventually enabled on this port.

To see what ports are currently on, and their status, as reported within NCBI, please refer to the following **Firewall Daemon Presence Check** page (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/fwd_check.cgi). Ports marked **INTERNAL** are for NCBI use only and may be inaccessible from your site without, however, affecting availability of any services that NCBI provides.

TROUBLESHOOTING: You can test if these ports are accessible from your host by just running, for example (see the "Ports to open" list above):

```
  telnet 130.14.29.112 5861
```

and entering a line of arbitrary text in the telnet session. If everything is fine, your TELNET session will look as follows (the line "test" is your input here):

```
| > telnet 130.14.29.112 5861
| Trying 130.14.29.112...
| Connected to 130.14.29.112.
| Escape character is '^]'.
| test
| NCBI Firewall Daemon:  Invalid ticket. Connection closed.
| Connection closed by foreign host.
```

There is also an auxiliary UNIX shell script **fwd_check.sh** (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/fwd_check.sh) to check all of the above addresses.

Note: Old NCBI clients used different application configuration settings and ports than listed above. If you need to support such clients, which are now obsolete, please contact info@ncbi.nlm.nih.gov for further information.

Parameters:

Input	
Remote DataBase	Select remote DB: **Non-Redundant** - All GenBank, EMBL and DDBJ Non-Redundant sequences (but no EST, STS, GSS, or phase 0, 1 or 2 HTGS sequences). WGS entries are also excluded. No longer "Non-Redundant". **EST** - Database for entries from Estimated Sequence Tags (EST) division of GenBank, EMBL and DDBJ. **Human EST** - H.Sapiens subset of Estimated Sequence Tags. **Mouse EST** - M.Musculus subset of Estimated Sequence Tags. **Other EST** - EST other than Human or Mouse. **GSS** - Genomic Survey Sequence, includes single-pass genomic data, exon-trapped sequences, and Alu PCR sequences. **HTGS** - Unfinished High Throughput Genomic Sequences: phases 0, 1 and 2. Finished, phase 3 HTG sequences are in NR. **Patented sequences (PAT)** - Nucleotides from the Patent division of GenBank. **Monthly Sequences (Month)** - All new or revised GenBank, EMBL and DDBJ sequences released updated in the last 30 days. **Alu repeats** - Select Alu repeats from REPBASE, suitable for masking Alu repeats from query sequences. **STS** - Database of GenBank, EMBL and DDBJ sequences from STS Division. **Chromosomic Sequences** - Complete genomes, complete chromosomes, or concatenated genomic contigs from NCBI Reference Sequence Project. **Vector fragments (UniVec)** - The UniVec non-redundant vector fragment sequences. **Whole Genome Shotguns (WGS)** - Whole Genome Shotgun sequence assembly. **Custom** - Specify the database of your interest.
Nucleotide Query sequence(s)	If the input file contains multiple sequences, BLAST will be run on each sequence in order, and the resulting output will contain concatenated BLAST reports.
Output	
Result	Designates an output file for the search results.
Format	Pairwise (Default) Query-anchored, showing identities

	Query-anchored, no identities Flat query-anchored, showing identities Flat query-anchored, no identities Query-anchored, no identities and blunt ends Flat query-anchored, no identities and blunt ends XML Blast output Tabular Tabular with comment lines ASN, text ASN, binary
Show GI's in deflines	Shows GenInfo Identifier (GI) numbers in definition lines. A GI is a unique numeric identifier assigned for a sequence in GenBank. A GI corresponds to an accession version pair.
Produce HTML output	Produces HTML output with [anchor] links from the summary at the top of the report to the alignments farther below. This option should be used only with the standard report format ("Pairwise (Default)").
Options	
MegaBlast search	Sets the blastn program to the megablast mode, which is optimized to find near identities very quickly.
Expectation value	Sets the threshold expectation value for keeping alignments. This is the E from the Karlin-Altschul equation that describes how often an alignment with a given score is expected to occur at random.
Filter query sequence	Filters the query sequence for low-complexity subsequences. The default setting is ON. Complexity filtering is generally a good idea, but it may break long HSPs into several smaller HSPs due to low-complexity segments. This can cause some alignments to fall below the significance threshold and be lost. To prevent this, either turn off filtering (not recommended) or use soft masking, in which the filter is used only in the word seeding phase, but not the extension phase. DUST with blastn, SEG with others.
Perform gapped alignment	Performs gapped alignment. Setting this to OFF invokes the older, ungapped style of alignment. You can't perform gapped alignments with tblastx, regardless of this setting.
Open Gap Cost	Initial penalty for opening a gap of length 0. -1 invokes the default behavior, and setting the parameter to zero is impossible, unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap costs for programs other than blastn depend on the scoring matrix.
Extend Gap Cost	The penalty for each gap character. Note that value -1 is synonymous with the default behavior for the "Open Gap Cost" parameter and, it's impossible to set value to zero unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap cost, for programs other than blastn, depends on the scoring matrix.
Nucleotide Mismatch Penalty	Sets the penalty for a nucleotide mismatch. Also see "Nucleotide Match Reward". The choice of [integer] for "Nucleotide Mismatch Penalty" and "Nucleotide Match Reward" are very important because they determine your target frequencies. The default values 1 for "Nucleotide Match Reward" and -3 for "Nucleotide Mismatch Penalty" are most effective for aligning sequences that are 99 percent identical.
Nucleotide Match	Sets the score of a nucleotide match. See also the "Nucleotide Mismatch

Reward	Penalty" parameter.
Number of DB Seqs to show descriptions	Sets the number of database sequences for which to show the one-line summary descriptions at the top of a BLAST report. You won't be warned if you exceed a value. Also see the "Number of Alignments to output" parameter.
Query strands	Chooses which strand of DNA-based queries is searched. **Top Strand** **Bottom Strand** **Both Strands**
Location on query sequence	The location on query sequence. This lets you limit the search to a subsequence of the query sequence. For example, to search just the letters from 21 to 50, set the parameter to following: "21,50" The alignments won't extend outside the specified region. In older versions of BLAST, this parameter set the size of the region under control of the "Best Hits Number" parameter.
Search Space Effective Length	Effective length of the search space. Use zero for the real size (Default).

NetBlastP

BLAST protein search program (net search).

Variant of the BlastP program intended for work with distant databases.
BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.
The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.
Reference:
Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.
Please, pay attention to following recommendations NCBI (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/firewall.html):
When first downloaded, your NCBI application runs in stand-alone mode, without access to the network. However, your program can also be configured to exchange information with the NCBI (GenBank) over the Internet. The network-aware mode of your application is identical to the stand-alone mode, but it contains some additional useful options.
Your application can only function in its network-aware mode if the computer on which it resides has a direct Internet connection. Electronic mail access to the Internet is insufficient. In general, if you can install and use a WWW-browser on your system, you should be able to install and use the network. Check with your system administrator or Internet provider if you are uncertain as to whether you have direct Internet connectivity.
To launch the configuration form, select Net Configure under the Misc menu in Sequin or Network Entrez, or the Options menu in Cn3D. If you are using blastcl3, you must run Sequin, Network Entrez, or Cn3D first to configure blastcl3. This is necessary because blastcl3 has no graphical user interface.
If you are not behind a firewall, set the **Connection** control to **Normal**. If you also have a Domain Name Server (DNS) available, you can now simply press **Accept**.
If DNS is not available, uncheck the **Domain Name Server** button. If you are behind a firewall, set the **Connection** control to **Firewall**. The **Proxy** box then becomes active. If you also use a

proxy server, type in its address. (If you have DNS, it will be of the form `www.myproxy.myuniversity.edu`. If you do not have DNS, you should use the numerical IP address of the form `127.45.23.6`.) Once you type something in the **Proxy** box, the **Port** box and **Transparent Proxy** button become active and can be filled in or changed as appropriate. (By default the **Transparent Proxy** button is off, indicating a CERN-like proxy.) Ask your network administrator for advice on the proper settings to use.

If you are in the United States, the default **Timeout** of 30 seconds should suffice. From foreign countries with poor Internet connection to the U.S., you can select up to 5 minutes as the timeout.

Finally, you will need to quit and restart your application in order for the network-aware settings to take effect.

If you are behind a firewall, it must be configured correctly to access NCBI services. Your network administrators may have done this already. If not, please have them read the section below.

The following section is intended for network administrators:

Using NCBI services from behind a security firewall requires opening ports in your firewall. The ports to open are:

```
Firewall Port        IP Address
--------------------------------
 5860..5870         130.14.29.112
    5845            130.14.22.12  (cannot be accessed from outside NCBI!)
```

If your firewall is not transparent, the firewall port number should be mapped to the same port number on the external host.

Port 5860 is usually not accessible by the public but reserved for NCBI internal purposes only. However, we recommend that it is kept open just as all other ports in the range in case the public access will be eventually enabled on this port.

To see what ports are currently on, and their status, as reported within NCBI, please refer to the following **Firewall Daemon Presence Check** page (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/fwd_check.cgi). Ports marked **INTERNAL** are for NCBI use only and may be inaccessible from your site without, however, affecting availability of any services that NCBI provides.

TROUBLESHOOTING: You can test if these ports are accessible from your host by just running, for example (see the "Ports to open" list above):

```
  telnet 130.14.29.112 5861
```

and entering a line of arbitrary text in the telnet session. If everything is fine, your TELNET session will look as follows (the line "test" is your input here):

```
| > telnet 130.14.29.112 5861
| Trying 130.14.29.112...
| Connected to 130.14.29.112.
| Escape character is '^]'.
| test
| NCBI Firewall Daemon:  Invalid ticket. Connection closed.
| Connection closed by foreign host.
```

There is also an auxiliary UNIX shell script **fwd_check.sh** (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/fwd_check.sh) to check all of the above addresses.

Note: Old NCBI clients used different application configuration settings and ports than listed above. If you need to support such clients, which are now obsolete, please contact info@ncbi.nlm.nih.gov for further information.

Parameters:

Input	
Remote DataBase	Remote DataBase selection: **Non-Redundant** - All Non-Redundant GenBank CDS translations, PDB, SwissProt, PIR and PRF. Non-Redundant.

	SwissProt DB - Last major release of the SWISS-PROT protein sequence database (no updates). **Patent Protein Sequence (PAT)** - Patent Protein Sequence database. **PDB Records** - Sequences derived from the 3-Dimensional structure records from PDB. **Monthly Sequences (Month)** - All new or revised GenBank CDS translations, PDB, SwissProt, PIR and PRF released in the last 30 days. **Custom** - Specify the database of your interest.
Nucleotide Query sequence(s)	If the input file contains multiple sequences, BLAST will be run on each sequence in order, and the resulting output will contain concatenated BLAST reports.
Believe the query defline.	Believe the query definition line.
Output	
Result	Designates an output file for the search results.
Format	Pairwise (Default) Query-anchored, showing identities Query-anchored, no identities Flat query-anchored, showing identities Flat query-anchored, no identities Query-anchored, no identities and blunt ends Flat query-anchored, no identities and blunt ends XML Blast output Tabular Tabular with comment lines ASN, text ASN, binary
Show GI's in deflines	Shows GenInfo Identifier (GI) numbers in definition lines. A GI is a unique numeric identifier assigned for a sequence in GenBank. A GI corresponds to an accession version pair.
Produce HTML output	Produces HTML output with [anchor] links from the summary at the top of the report to the alignments farther below. This option should be used only with the standard report format ("Pairwise (Default)").
Options	
Expectation value	Sets the threshold expectation value for keeping alignments. This is the E from the Karlin-Altschul equation that describes how often an alignment with a given score is expected to occur at random.
Filter query sequence	Filters the query sequence for low-complexity subsequences. The default setting is ON. Complexity filtering is generally a good idea, but it may break long HSPs into several smaller HSPs due to low-complexity segments. This can cause some alignments to fall below the significance threshold and be lost. To prevent this, either turn off filtering (not recommended) or use soft masking, in which the filter is used only in the word seeding phase, but not the extension phase. DUST with blastn, SEG with others.
Perform gapped alignment	Performs gapped alignment. Setting this to OFF invokes the older, ungapped style of alignment. You can't perform gapped alignments with tblastx, regardless of this setting.
Open Gap Cost	Initial penalty for opening a gap of length 0. -1 invokes the default behavior,

	and setting the parameter to zero is impossible, unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap costs for programs other than blastn depend on the scoring matrix.
Extend Gap Cost	The penalty for each gap character. Note that value -1 is synonymous with the default behavior for the "Open Gap Cost" parameter and, it's impossible to set value to zero unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap cost, for programs other than blastn, depends on the scoring matrix.
Matrix	Designates a protein similarity matrix. This is used in all BLAST programs except blastn. Matrices are sought in the following order: in the local directory, in the location specified in the .ncbirc file, in a local data directory, and finally, in the BLASTMAT environment variable (only on Unix systems). Other matrices included in the standard distribution include BLOSUM45, BLOSUM80, PAM30, and PAM70. You can use custom matrix files, but it requires modifying the source code and defining the new matrix with all of its associated statistics for different affine gap combinations and recompiling the binary. Using these custom files isn't recommended because it requires the arduous task of calculating gapped values for lambda and maintaining a derivative branch of the source code.
Query strands	Chooses which strand of DNA-based queries is searched. **Top Strand** **Bottom Strand** **Both Strands**
Location on query sequence	The location on query sequence. This lets you limit the search to a subsequence of the query sequence. For example, to search just the letters from 21 to 50, set the parameter to following: "21,50". The alignments won't extend outside the specified region. In older versions of BLAST, this parameter set the size of the region under control of the "Best Hits Number" parameter.
Search Space Effective Length	Effective length of the search space. Use zero for the real size (Default).
Lower Case Filtering	Use lower case filtering of FASTA sequence.
Ungapped Extension X dropoff value	X dropoff value for ungapped extensions in bits; Zero invokes default behavior; blastn 20, megablast 10, all others 7.
Final Gapped Alignment X dropoff value	X dropoff value for final gapped alignment in bits; Sets the X3 dropoff value (in bits) for extensions but is bounded by the value for X2. Zero invokes default behavior; blastn/megablast 50, tblastx 0, all others 25.

NetBlastX

BLASTX is generally used to find protein coding genes in genomic DNA or to identify proteins encoded by transcripts.
Most proteins are related to other proteins. This makes BLASTX a very powerful gene-finding tool. As protein databases become larger and more diverse, BLASTX becomes even more useful because it can identify more and more genes.
Net-BlastX is a variant of the BlastX program intended for work with distant databases. BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide

or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.

The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.

Reference:

Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Please, pay attention to following recommendations NCBI (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/firewall.html):

When first downloaded, your NCBI application runs in stand-alone mode, without access to the network. However, your program can also be configured to exchange information with the NCBI (GenBank) over the Internet. The network-aware mode of your application is identical to the stand-alone mode, but it contains some additional useful options.

Your application can only function in its network-aware mode if the computer on which it resides has a direct Internet connection. Electronic mail access to the Internet is insufficient. In general, if you can install and use a WWW-browser on your system, you should be able to install and use the network. Check with your system administrator or Internet provider if you are uncertain as to whether you have direct Internet connectivity.

To launch the configuration form, select Net Configure under the Misc menu in Sequin or Network Entrez, or the Options menu in Cn3D. If you are using blastcl3, you must run Sequin, Network Entrez, or Cn3D first to configure blastcl3. This is necessary because blastcl3 has no graphical user interface.

If you are not behind a firewall, set the **Connection** control to **Normal**. If you also have a Domain Name Server (DNS) available, you can now simply press **Accept**.

If DNS is not available, uncheck the **Domain Name Server** button. If you are behind a firewall, set the **Connection** control to **Firewall**. The **Proxy** box then becomes active. If you also use a proxy server, type in its address. (If you have DNS, it will be of the form `www.myproxy.myuniversity.edu`. If you do not have DNS, you should use the numerical IP address of the form `127.45.23.6`.) Once you type something in the **Proxy** box, the **Port** box and **Transparent Proxy** button become active and can be filled in or changed as appropriate. (By default the **Transparent Proxy** button is off, indicating a CERN-like proxy.) Ask your network administrator for advice on the proper settings to use.

If you are in the United States, the default **Timeout** of 30 seconds should suffice. From foreign countries with poor Internet connection to the U.S., you can select up to 5 minutes as the timeout.

Finally, you will need to quit and restart your application in order for the network-aware settings to take effect.

If you are behind a firewall, it must be configured correctly to access NCBI services. Your network administrators may have done this already. If not, please have them read the section below.

The following section is intended for network administrators:

Using NCBI services from behind a security firewall requires opening ports in your firewall. The ports to open are:

```
Firewall Port        IP Address
--------------------------------
 5860..5870         130.14.29.112
    5845            130.14.22.12   (cannot be accessed from outside NCBI!)
```

If your firewall is not transparent, the firewall port number should be mapped to the same port number on the external host.

Port 5860 is usually not accessible by the public but reserved for NCBI internal purposes only. However, we recommend that it is kept open just as all other ports in the range in case the public access will be eventually enabled on this port.

To see what ports are currently on, and their status, as reported within NCBI, please refer to the following **Firewall Daemon Presence Check** page (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/fwd_check.cgi). Ports marked **INTERNAL** are for NCBI use only and may be inaccessible from your site without, however, affecting availability of any services that NCBI provides.

TROUBLESHOOTING: You can test if these ports are accessible from your host by just running, for example (see the "Ports to open" list above):

```
  telnet 130.14.29.112 5861
```

and entering a line of arbitrary text in the telnet session. If everything is fine, your TELNET session will look as follows (the line "test" is your input here):

```
| > telnet 130.14.29.112 5861
| Trying 130.14.29.112...
| Connected to 130.14.29.112.
| Escape character is '^]'.
| test
| NCBI Firewall Daemon:  Invalid ticket. Connection closed.
| Connection closed by foreign host.
```

There is also an auxiliary UNIX shell script **fwd_check.sh** (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/fwd_check.sh) to check all of the above addresses.

Note: Old NCBI clients used different application configuration settings and ports than listed above. If you need to support such clients, which are now obsolete, please contact info@ncbi.nlm.nih.gov for further information.

Parameters:

Input	
Remote DataBase	Remote DataBase selection: **Non-Redundant** - All Non-Redundant GenBank CDS translations, PDB, SwissProt, PIR and PRF. Non-Redundant. **SwissProt DB** - Last major release of the SWISS-PROT protein sequence database (no updates). **Patent Protein Sequence (PAT)** - Patent Protein Sequence database. **PDB Records** - Sequences derived from the 3-Dimensional structure records from PDB. **Monthly Sequences (Month)** - All new or revised GenBank CDS translations, PDB, SwissProt, PIR and PRF released in the last 30 days. **Custom** - Specify the database of your interest.
Nucleotide Query sequence(s)	If the input file contains multiple sequences, BLAST will be run on each sequence in order, and the resulting output will contain concatenated BLAST reports.
Believe the query defline.	Believe the query definition line.
Output	
Result	Designates an output file for the search results.
Format	Pairwise (Default) Query-anchored, showing identities Query-anchored, no identities Flat query-anchored, showing identities Flat query-anchored, no identities Query-anchored, no identities and blunt ends Flat query-anchored, no identities and blunt ends XML Blast output

	Tabular Tabular with comment lines ASN, text ASN, binary
Show GI's in deflines	Shows GenInfo Identifier (GI) numbers in definition lines. A GI is a unique numeric identifier assigned for a sequence in GenBank. A GI corresponds to an accession version pair.
Produce HTML output	Produces HTML output with [anchor] links from the summary at the top of the report to the alignments farther below. This option should be used only with the standard report format ("Pairwise (Default)").
Options	
Expectation value	Sets the threshold expectation value for keeping alignments. This is the E from the Karlin-Altschul equation that describes how often an alignment with a given score is expected to occur at random.
Filter query sequence	Filters the query sequence for low-complexity subsequences. The default setting is ON. Complexity filtering is generally a good idea, but it may break long HSPs into several smaller HSPs due to low-complexity segments. This can cause some alignments to fall below the significance threshold and be lost. To prevent this, either turn off filtering (not recommended) or use soft masking, in which the filter is used only in the word seeding phase, but not the extension phase. DUST with blastn, SEG with others.
Perform gapped alignment	Performs gapped alignment. Setting this to OFF invokes the older, ungapped style of alignment. You can't perform gapped alignments with tblastx, regardless of this setting.
Open Gap Cost	Initial penalty for opening a gap of length 0. -1 invokes the default behavior, and setting the parameter to zero is impossible, unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap costs for programs other than blastn depend on the scoring matrix.
Extend Gap Cost	The penalty for each gap character. Note that value -1 is synonymous with the default behavior for the "Open Gap Cost" parameter and, it's impossible to set value to zero unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap cost, for programs other than blastn, depends on the scoring matrix.
Translation table	Select translation table.
Matrix	Designates a protein similarity matrix. This is used in all BLAST programs except blastn. Matrices are sought in the following order: in the local directory, in the location specified in the .ncbirc file, in a local data directory, and finally, in the BLASTMAT environment variable (only on Unix systems). Other matrices included in the standard distribution include BLOSUM45, BLOSUM80, PAM30, and PAM70. You can use custom matrix files, but it requires modifying the source code and defining the new matrix with all of its associated statistics for different affine gap combinations and recompiling the binary. Using these custom files isn't recommended because it requires the arduous task of calculating gapped values for lambda and maintaining a derivative branch of the source code.
Query strands	Chooses which strand of DNA-based queries is searched.

	Top Strand **Bottom Strand** **Both Strands**
Location on query sequence	The location on query sequence. This lets you limit the search to a subsequence of the query sequence. For example, to search just the letters from 21 to 50, set the parameter to following: "21,50". The alignments won't extend outside the specified region. In older versions of BLAST, this parameter set the size of the region under control of the "Best Hits Number" parameter.
Search Space Effective Length	Effective length of the search space. Use zero for the real size (Default).
Lower Case Filtering	Use lower case filtering of FASTA sequence.
Ungapped Extension X dropoff value	X dropoff value for ungapped extensions in bits; Zero invokes default behavior; blastn 20, megablast 10, all others 7.
Final Gapped Alignment X dropoff value	X dropoff value for final gapped alignment in bits; Sets the X3 dropoff value (in bits) for extensions but is bounded by the value for X2. Zero invokes default behavior; blastn/megablast 50, tblastx 0, all others 25.
Frame shift penalty	Sets the frame shift penalty for the Out Of Frame (OOF) algorithm of blastx. When the parameter is set, it invokes the OOF mode of BLAST, which lets alignments proceed across reading frames. The expect values calculated from OOF blastx are only approximate, and BLAST issues the following warning when OOF is invoked: [NULL_Caption] WARNING: test500: Out-of-frame option selected, Expect values are only approximate and calculated not assuming out-of-frame alignments The out-of-frame alignments are signified by slashes that indicate the +1(/),+2(//), -1(\), and -2(\\) frameshifts. The following is a sample OOF alignment:

```
Query: 23  PLIRNSL/YCINC\\A//QSIIRAHVKGPYLTRWVVNC/E\TCSKGYAKTPGASTDLLLL 160
           PLIRNSL YCINC     QSIIRAHVKGPYLTRWVVNC   TCSKGYAKTPGASTDLLLL
Sbjct: 1   PLIRNSL YCINC  X  QSIIRAHVKGPYLTRWVVNC X TCSKGYAKTPGASTDLLLL 53
Query: 161 YKTRNSLTSASSLSPVRSQRMI/N\SFPRFQGHLVVSG/S\SAHNR/FS\FNRDSPRGSG 322
           YKTRNSLTSASSLSPVRSQRMI   SFPRFQGHLVVSG   SAHNR F  FNRDSPRGSG
Sbjct: 54  YKTRNSLTSASSLSPVRSQRMI X SFPRFQGHLVVSG X SAHNR FX FNRDSPRGSG 107
Query: 323 SYCSREPMGQIKIRRTHTDDKLFR/ND\SRHTRAGDGLNI//TLA\\RDPSFLSRVYNAN 484
           SYCSREPMGQIKIRRTHTDDKLFR    SRHTRAGDGLNI   L   RDPSFLSRVYNAN
Sbjct: 108 SYCSREPMGQIKIRRTHTDDKLFR XX SRHTRAGDGLNI  XLX  RDPSFLSRVYNAN 161
Query: 485 SYLHI 499
           SYLHI
Sbjct: 162 SYLHI 166
```

Net-tBlastN

TBLASTN commonly maps a protein to a genome or searches EST databases for related proteins not yet in the protein databases.
Net-tBlastN is a variant of the tBlastN program intended for work with distant databases. BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.

The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.

Reference:

Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Please, pay attention to following recommendations NCBI (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/firewall.html):

When first downloaded, your NCBI application runs in stand-alone mode, without access to the network. However, your program can also be configured to exchange information with the NCBI (GenBank) over the Internet. The network-aware mode of your application is identical to the stand-alone mode, but it contains some additional useful options.

Your application can only function in its network-aware mode if the computer on which it resides has a direct Internet connection. Electronic mail access to the Internet is insufficient. In general, if you can install and use a WWW-browser on your system, you should be able to install and use the network. Check with your system administrator or Internet provider if you are uncertain as to whether you have direct Internet connectivity.

To launch the configuration form, select Net Configure under the Misc menu in Sequin or Network Entrez, or the Options menu in Cn3D. If you are using blastcl3, you must run Sequin, Network Entrez, or Cn3D first to configure blastcl3. This is necessary because blastcl3 has no graphical user interface.

If you are not behind a firewall, set the **Connection** control to **Normal**. If you also have a Domain Name Server (DNS) available, you can now simply press **Accept**.

If DNS is not available, uncheck the **Domain Name Server** button. If you are behind a firewall, set the **Connection** control to **Firewall**. The **Proxy** box then becomes active. If you also use a proxy server, type in its address. (If you have DNS, it will be of the form `www.myproxy.myuniversity.edu`. If you do not have DNS, you should use the numerical IP address of the form `127.45.23.6`.) Once you type something in the **Proxy** box, the **Port** box and **Transparent Proxy** button become active and can be filled in or changed as appropriate. (By default the **Transparent Proxy** button is off, indicating a CERN-like proxy.) Ask your network administrator for advice on the proper settings to use.

If you are in the United States, the default **Timeout** of 30 seconds should suffice. From foreign countries with poor Internet connection to the U.S., you can select up to 5 minutes as the timeout.

Finally, you will need to quit and restart your application in order for the network-aware settings to take effect.

If you are behind a firewall, it must be configured correctly to access NCBI services. Your network administrators may have done this already. If not, please have them read the section below.

The following section is intended for network administrators:

Using NCBI services from behind a security firewall requires opening ports in your firewall. The ports to open are:

```
Firewall Port        IP Address
--------------------------------
 5860..5870         130.14.29.112
    5845            130.14.22.12   (cannot be accessed from outside NCBI!)
```

If your firewall is not transparent, the firewall port number should be mapped to the same port number on the external host.

Port 5860 is usually not accessible by the public but reserved for NCBI internal purposes only. However, we recommend that it is kept open just as all other ports in the range in case the public access will be eventually enabled on this port.

To see what ports are currently on, and their status, as reported within NCBI, please refer to the following **Firewall Daemon Presence Check** page (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/fwd_check.cgi). Ports marked **INTERNAL** are for NCBI use only and may be inaccessible from your site without, however, affecting availability of any services that NCBI provides.

TROUBLESHOOTING: You can test if these ports are accessible from your host by just running, for example (see the "Ports to open" list above):

```
  telnet 130.14.29.112 5861
```

and entering a line of arbitrary text in the telnet session. If everything is fine, your TELNET session will look as follows (the line "test" is your input here):

```
| > telnet 130.14.29.112 5861
| Trying 130.14.29.112...
| Connected to 130.14.29.112.
| Escape character is '^]'.
| test
| NCBI Firewall Daemon:  Invalid ticket. Connection closed.
| Connection closed by foreign host.
```

There is also an auxiliary UNIX shell script **fwd_check.sh** (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/fwd_check.sh) to check all of the above addresses.

Note: Old NCBI clients used different application configuration settings and ports than listed above. If you need to support such clients, which are now obsolete, please contact info@ncbi.nlm.nih.gov for further information.

Parameters:

Input	
Remote DataBase	Select remote DB: **Non-Redundant** - All GenBank, EMBL and DDBJ Non-Redundant sequences (but no EST, STS, GSS, or phase 0, 1 or 2 HTGS sequences). WGS entries are also excluded. No longer "Non-Redundant". **EST** - Database for entries from Estimated Sequence Tags (EST) division of GenBank, EMBL and DDBJ. **Human EST** - H.Sapiens subset of Estimated Sequence Tags. **Mouse EST** - M.Musculus subset of Estimated Sequence Tags. **Other EST** - EST other than Human or Mouse. **GSS** - Genomic Survey Sequence, includes single-pass genomic data, exon-trapped sequences, and Alu PCR sequences. **HTGS** - Unfinished High Throughput Genomic Sequences: phases 0, 1 and 2. Finished, phase 3 HTG sequences are in NR. **Patented sequences (PAT)** - Nucleotides from the Patent division of GenBank. **Monthly Sequences (Month)** - All new or revised GenBank, EMBL and DDBJ sequences released updated in the last 30 days. **Alu repeats** - Select Alu repeats from REPBASE, suitable for masking Alu repeats from query sequences. **STS** - Database of GenBank, EMBL and DDBJ sequences from STS Division. **Chromosomic Sequences** - Complete genomes, complete chromosomes, or concatenated genomic contigs from NCBI Reference Sequence Project. **Vector fragments (UniVec)** - The UniVec non-redundant vector fragment sequences. **Whole Genome Shotguns (WGS)** - Whole Genome Shotgun sequence assembly. **Custom** - Specify the database of your interest.
Protein Query sequence(s)	If the input file contains multiple sequences, BLAST will be run on each sequence in order, and the resulting output will contain concatenated BLAST

	reports.
Believe the query defline.	Believe the query definition line.
Output	
Result	Designates an output file for the search results.
Format	Pairwise (Default) Query-anchored, showing identities Query-anchored, no identities Flat query-anchored, showing identities Flat query-anchored, no identities Query-anchored, no identities and blunt ends Flat query-anchored, no identities and blunt ends XML Blast output Tabular Tabular with comment lines ASN, text ASN, binary
Show GI's in deflines	Shows GenInfo Identifier (GI) numbers in definition lines. A GI is a unique numeric identifier assigned for a sequence in GenBank. A GI corresponds to an accession version pair.
Produce HTML output	Produces HTML output with [anchor] links from the summary at the top of the report to the alignments farther below. This option should be used only with the standard report format ("Pairwise (Default)").
Options	
Expectation value	Sets the threshold expectation value for keeping alignments. This is the E from the Karlin-Altschul equation that describes how often an alignment with a given score is expected to occur at random.
Filter query sequence	Filters the query sequence for low-complexity subsequences. The default setting is ON. Complexity filtering is generally a good idea, but it may break long HSPs into several smaller HSPs due to low-complexity segments. This can cause some alignments to fall below the significance threshold and be lost. To prevent this, either turn off filtering (not recommended) or use soft masking, in which the filter is used only in the word seeding phase, but not the extension phase. DUST with blastn, SEG with others.
Perform gapped alignment	Performs gapped alignment. Setting this to OFF invokes the older, ungapped style of alignment. You can't perform gapped alignments with tblastx, regardless of this setting.
Open Gap Cost	Initial penalty for opening a gap of length 0. -1 invokes the default behavior, and setting the parameter to zero is impossible, unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap costs for programs other than blastn depend on the scoring matrix.
Extend Gap Cost	The penalty for each gap character. Note that value -1 is synonymous with the default behavior for the "Open Gap Cost" parameter and, it's impossible to set value to zero unless the "Perform gapped alignment" option is set to NO, which turns gapping off. The default gap cost, for programs other than blastn, depends on the scoring matrix.
DB Genetic code	The genetic code to use for translation of the database nucleotide sequence.

	See http://www.ncbi.nlm.nih.gov/htbin-post/Taxonomy for updates
Matrix	Designates a protein similarity matrix. This is used in all BLAST programs except blastn. Matrices are sought in the following order: in the local directory, in the location specified in the .ncbirc file, in a local data directory, and finally, in the BLASTMAT environment variable (only on Unix systems). Other matrices included in the standard distribution include BLOSUM45, BLOSUM80, PAM30, and PAM70. You can use custom matrix files, but it requires modifying the source code and defining the new matrix with all of its associated statistics for different affine gap combinations and recompiling the binary. Using these custom files isn't recommended because it requires the arduous task of calculating gapped values for lambda and maintaining a derivative branch of the source code.
Location on query sequence	The location on query sequence. This lets you limit the search to a subsequence of the query sequence. For example, to search just the letters from 21 to 50, set the parameter to following: "21,50" The alignments won't extend outside the specified region. In older versions of BLAST, this parameter set the size of the region under control of the "Best Hits Number" parameter.
Search Space Effective Length	Effective length of the search space. Use zero for the real size (Default).
Composition-based statistics	Use composition-based statistics for tblastn. For programs other than tblastn, must be absent (Default). Possible choices:. 1. Composition-based statistics as in NAR 29:2994-3005, 2001. 2. Composition-based score adjustment as in Bioinformatics 21:902-911, 2005, conditioned on sequence properties. 3. Composition-based score adjustment as in Bioinformatics 21:902-911, 2005, unconditionally.

Net-tBlastX

TBLASTX is a powerful gene-prediction tool for genomes that are appropriately diverged. TBLASTX translates both strands of the query and nucleotide database sequences in three frames on each strand, and examine all pairwise combinations to find similarities at the amino acid level.

Net-tBlastX is a variant of the tBlastX program intended for work with distant databases.

!NOTE! Because this program involves more computation than the others, it is not recommended to search of the Non-redundant (nr) database.

BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.

The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.

Reference:

Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Please, pay attention to following recommendations NCBI (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/firewall.html):

When first downloaded, your NCBI application runs in stand-alone mode, without access to the network. However, your program can also be configured to exchange information with the NCBI (GenBank) over the Internet. The network-aware mode of your application is identical to the stand-alone mode, but it contains some additional useful options.

Your application can only function in its network-aware mode if the computer on which it resides has a direct Internet connection. Electronic mail access to the Internet is insufficient. In general, if you can install and use a WWW-browser on your system, you should be able to install and use the network. Check with your system administrator or Internet provider if you are uncertain as to whether you have direct Internet connectivity.

To launch the configuration form, select Net Configure under the Misc menu in Sequin or Network Entrez, or the Options menu in Cn3D. If you are using blastcl3, you must run Sequin, Network Entrez, or Cn3D first to configure blastcl3. This is necessary because blastcl3 has no graphical user interface.

If you are not behind a firewall, set the **Connection** control to **Normal**. If you also have a Domain Name Server (DNS) available, you can now simply press **Accept**.

If DNS is not available, uncheck the **Domain Name Server** button. If you are behind a firewall, set the **Connection** control to **Firewall**. The **Proxy** box then becomes active. If you also use a proxy server, type in its address. (If you have DNS, it will be of the form `www.myproxy.myuniversity.edu`. If you do not have DNS, you should use the numerical IP address of the form `127.45.23.6`.) Once you type something in the **Proxy** box, the **Port** box and **Transparent Proxy** button become active and can be filled in or changed as appropriate. (By default the **Transparent Proxy** button is off, indicating a CERN-like proxy.) Ask your network administrator for advice on the proper settings to use.

If you are in the United States, the default **Timeout** of 30 seconds should suffice. From foreign countries with poor Internet connection to the U.S., you can select up to 5 minutes as the timeout.

Finally, you will need to quit and restart your application in order for the network-aware settings to take effect.

If you are behind a firewall, it must be configured correctly to access NCBI services. Your network administrators may have done this already. If not, please have them read the section below.

The following section is intended for network administrators:

Using NCBI services from behind a security firewall requires opening ports in your firewall. The ports to open are:

```
Firewall Port        IP Address
--------------------------------
 5860..5870         130.14.29.112
    5845            130.14.22.12  (cannot be accessed from outside NCBI!)
```

If your firewall is not transparent, the firewall port number should be mapped to the same port number on the external host.

Port 5860 is usually not accessible by the public but reserved for NCBI internal purposes only. However, we recommend that it is kept open just as all other ports in the range in case the public access will be eventually enabled on this port.

To see what ports are currently on, and their status, as reported within NCBI, please refer to the following **Firewall Daemon Presence Check** page (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/fwd_check.cgi). Ports marked **INTERNAL** are for NCBI use only and may be inaccessible from your site without, however, affecting availability of any services that NCBI provides.

TROUBLESHOOTING: You can test if these ports are accessible from your host by just running, for example (see the "Ports to open" list above):

```
  telnet 130.14.29.112 5861
```

and entering a line of arbitrary text in the telnet session. If everything is fine, your TELNET session will look as follows (the line "test" is your input here):

```
| > telnet 130.14.29.112 5861
| Trying 130.14.29.112...
| Connected to 130.14.29.112.
| Escape character is '^]'.
| test
| NCBI Firewall Daemon:  Invalid ticket. Connection closed.
| Connection closed by foreign host.
```

There is also an auxiliary UNIX shell script **fwd_check.sh** (http://www.ncbi.nlm.nih.gov/IEB/ToolBox/NETWORK/fwd_check.sh) to check all of the above addresses.

Note: Old NCBI clients used different application configuration settings and ports than listed above. If you need to support such clients, which are now obsolete, please contact info@ncbi.nlm.nih.gov for further information.

Parameters:

Input	
Remote DataBase	Select remote DB: **Non-Redundant** - All GenBank, EMBL and DDBJ Non-Redundant sequences (but no EST, STS, GSS, or phase 0, 1 or 2 HTGS sequences). WGS entries are also excluded. No longer "Non-Redundant". **EST** - Database for entries from Estimated Sequence Tags (EST) division of GenBank, EMBL and DDBJ. **Human EST** - H.Sapiens subset of Estimated Sequence Tags. **Mouse EST** - M.Musculus subset of Estimated Sequence Tags. **Other EST** - EST other than Human or Mouse. **GSS** - Genomic Survey Sequence, includes single-pass genomic data, exon-trapped sequences, and Alu PCR sequences. **HTGS** - Unfinished High Throughput Genomic Sequences: phases 0, 1 and 2. Finished, phase 3 HTG sequences are in NR. **Patented sequences (PAT)** - Nucleotides from the Patent division of GenBank. **Monthly Sequences (Month)** - All new or revised GenBank, EMBL and DDBJ sequences released updated in the last 30 days. **Alu repeats** - Select Alu repeats from REPBASE, suitable for masking Alu repeats from query sequences. **STS** - Database of GenBank, EMBL and DDBJ sequences from STS Division. **Chromosomic Sequences** - Complete genomes, complete chromosomes, or concatenated genomic contigs from NCBI Reference Sequence Project. **Vector fragments (UniVec)** - The UniVec non-redundant vector fragment sequences. **Whole Genome Shotguns (WGS)** - Whole Genome Shotgun sequence assembly. **Custom** - Specify the database of your interest.
Nucleotide Query sequence(s)	If the input file contains multiple sequences, BLAST will be run on each sequence in order, and the resulting output will contain concatenated BLAST reports.
Believe the query defline.	Believe the query definition line.
Output	
Result	Designates an output file for the search results.
Format	Pairwise (Default) Query-anchored, showing identities Query-anchored, no identities Flat query-anchored, showing identities Flat query-anchored, no identities

	Query-anchored, no identities and blunt ends Flat query-anchored, no identities and blunt ends XML Blast output Tabular Tabular with comment lines ASN, text ASN, binary
Show GI's in deflines	Shows GenInfo Identifier (GI) numbers in definition lines. A GI is a unique numeric identifier assigned for a sequence in GenBank. A GI corresponds to an accession version pair.
Produce HTML output	Produces HTML output with [anchor] links from the summary at the top of the report to the alignments farther below. This option should be used only with the standard report format ("Pairwise (Default)").
Options	
Expectation value	Sets the threshold expectation value for keeping alignments. This is the E from the Karlin-Altschul equation that describes how often an alignment with a given score is expected to occur at random.
Filter query sequence	Filters the query sequence for low-complexity subsequences. The default setting is ON. Complexity filtering is generally a good idea, but it may break long HSPs into several smaller HSPs due to low-complexity segments. This can cause some alignments to fall below the significance threshold and be lost. To prevent this, either turn off filtering (not recommended) or use soft masking, in which the filter is used only in the word seeding phase, but not the extension phase. DUST with blastn, SEG with others.
Translation table	Select translation table.
DB Genetic code	The genetic code to use for translation of the database nucleotide sequence. See http://www.ncbi.nlm.nih.gov/htbin-post/Taxonomy for updates
Matrix	Designates a protein similarity matrix. This is used in all BLAST programs except blastn. Matrices are sought in the following order: in the local directory, in the location specified in the .ncbirc file, in a local data directory, and finally, in the BLASTMAT environment variable (only on Unix systems). Other matrices included in the standard distribution include BLOSUM45, BLOSUM80, PAM30, and PAM70. You can use custom matrix files, but it requires modifying the source code and defining the new matrix with all of its associated statistics for different affine gap combinations and recompiling the binary. Using these custom files isn't recommended because it requires the arduous task of calculating gapped values for lambda and maintaining a derivative branch of the source code.
Query strands	Chooses which strand of DNA-based queries is searched. **Top Strand** **Bottom Strand** **Both Strands**
Location on query sequence	The location on query sequence. This lets you limit the search to a subsequence of the query sequence. For example, to search just the letters from 21 to 50, set the parameter to following: "21,50".

	The alignments won't extend outside the specified region. In older versions of BLAST, this parameter set the size of the region under control of the "Best Hits Number" parameter.
Search Space Effective Length	Effective length of the search space. Use zero for the real size (Default).

PSI-Blast

The blastpgp program can do an iterative search in which sequences found in one round of searching are used to build a score model for the next round of searching.

The program aligns sequence (input file) on the base prepared by program FormatDB.

BLAST is a service of the National Center for Biotechnology Information (NCBI). A nucleotide or protein sequence sent to the BLAST server is compared against databases at the NCBI and a summary of matches is returned to the user.

The www BLAST server can be accessed through the home page of the NCBI at www.ncbi.nlm.nih.gov. Stand-alone BLAST binaries can be obtained from the NCBI FTP site.

The blastpgp program can do an iterative search in which sequences found in one round of searching are used to build a score model for the next round of searching. In this usage, the program is called Position-Specific Iterated BLAST, or PSI-BLAST. As explained in the accompanying paper, the BLAST algorithm is not tied to a specific score matrix. Traditionally, it has been implemented using an AxA substitution matrix where A is the alphabet size. PSI-BLAST instead uses a QxA matrix, where Q is the length of the query sequence; at each position the cost of a letter depends on the position w.r.t. the query and the letter in the subject sequence.

The position-specific matrix for round i+1 is built from a constrained multiple alignment among the query and the sequences found with sufficiently low e-value in round i. The top part of the output for each round distinguishes the sequences into: sequences found previously and used in the score model, and sequences not used in the score model. The output currently includes lots of diagnostics requested by users at NCBI. To skip quickly from the output of one round to the next, search for the string "producing", which is part of the header for each round and likely does not appear elsewhere in the output. PSI-BLAST "converges" and stops if all sequences found at round i+1 below the e-value threshold were already in the model at the beginning of the round.

Users who also develop their own sequence analysis software may wish to develop their own scoring systems. For this purpose the code in posit.c that writes out the checkpoint can be easily adapated to write out scoring systems derived by other algorithms in such a way that PSI-BLAST can read the files in later.

The checkpoint structure is general in the sense that it can handle any position-specific matrix that fits in the Karlin-Altschul statistical framework for BLAST scoring.

References

Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Parameters:

Input	
Sequence	Input file
Blast DB	Blast DB file
Hit data	Hit File for PHI-BLAST
Alignment data	Input Alignment File for PSI-BLAST Restart
Output	
Output file	Output file

Options	
Program name	Select search program: **blastpgp** **patmatchp** **patmatch** **patseedp** **patseed** **patternp** **pattern** **seedp** **seed**
Expectation value	Expectation value default = 10.0
Maximum number of rounds	The maximum number of rounds (default 1; i.e., regular BLAST)
Constant	The "constant" used in the pseudocount formula specified in the paper (default 10)

Net Data Access

Get PDB ID

The program performs retrieving PDB Identifiers from file with BlastP alignment
Parameters:

Input	
Blast Alignment File	File with results of BlastP protein aligning.
Output	
Result	Name of the output file.
Options	
Homology threshold	Specifying this parameter, user can discard results with homology percentage lower than set value.

NCBI-Expression

The program performs net access to NCBI databases.
Parameters:

Input	
Data Identifier(s)	List of Accession Numbers (use comma as a separator), can be used with Identifier(s) list .
Identifier(s) list	File with list of Accesion Numbers - list of values - each AC in new line.
Output	
Result file (CEL)	Name of the output file with data in Affymetrix CEL data format. The CEL file stores the results of the intensity calculations on the pixel values on the chip.
Result file (CHP)	Name of the output file with the set of expression data in Affymetrix CHP data format.
Result file (EXP)	Name of the output Affymetrix experiment description file.
Options	
Proxy settings	Proxy settings (protocol, login, password, host, port - ask your system administrator about this options)

NCBI-Genbank

The program performs net access to NCBI databases.
Parameters:

Input	
Data Identifier(s)	List of Accession Numbers (use comma as a separator), can be used with Identifier(s) list .
Identifier(s) list	File with list of Accesion Numbers - list of values - each AC in new line.
Output	
Result file	Name of the output file.
Options	
Proxy settings	Proxy settings (protocol, login, password, host, port - ask your system administrator about this options)

NCBI-Nucleic

The program performs net access to NCBI databases.

Parameters:

Input	
Data Identifier(s)	List of Accession Numbers (use comma as a separator), can be used with Identifier(s) list .
Identifier(s) list	File with list of Accesion Numbers - list of values - each AC in new line.
Output	
Result file	Name of the output file.
Options	
Proxy settings	Proxy settings (protocol, login, password, host, port - ask your system administrator about this options)

NCBI-PDB

The program performs net access to NCBI databases.

Input	
Data Identifier(s)	Accesion Number.
Output	
Result file	Name of the output file.
Options	
Proxy settings	Proxy settings (protocol, login, password, host, port - ask your system administrator about this options)

Parameters:

NCBI-Protein

The program performs net access to NCBI databases.

Parameters:

Input	
Data Identifier(s)	List of Accession Numbers (use comma as a separator), can be used with Identifier(s) list .
Identifier(s) list	File with list of Accesion Numbers - list of values - each AC in new line.
Output	
Result file	Name of the output file.
Options	
Proxy settings	Proxy settings (protocol, login, password, host, port - ask your system administrator about this options)

Promoter/Regulation

CPGFinder

The program is intended to search for CpG islands in sequences.
Output example:

```
Search parameters:  len: 200   %GC: 50.0   CpG number: 0   P(CpG)/exp: 0.600
extend island: no   A: 21   B: -2
Locus name:  9003..16734 note="CpG_island (%GC=65.4, o/e=0.70, #CpGs=577)"
Locus reference:   expected P(CpG): 0.086   length: 25020
    20.1%(a)  29.9%(c)  28.6%(g)  21.4%(t)   0.0%(other)

                                       FOUND 4 ISLANDS
  #     start      end   chain   CpG    %CG     CG/GC    P(CpG)/exp      P(CpG)    len
  1      9192    10496     +     161   73.0     0.847   0.927( 1.44)    0.123    1305
  2     11147    11939     +      87   69.2     0.821   0.917( 1.28)    0.110     793
  3     15957    16374     +      57   79.4     0.781   0.871( 1.60)    0.137     418
  4     14689    15091     +      49   74.2     0.817   0.887( 1.42)    0.122     403
```

Parameters:

Input	
Sequence	Input file - nucleotide sequence in FASTA-format
Output	
Result	Name of the output file
Options	
Minimal length of island	Searching CpG islands with a length (bp) not less than specified in the field.
Minimal percent G and C	Searching CpG islands with a composition not less than specified in the field.
Minimal GC ratio	The minimal ratio of the observed to expected frequency of CpG dinucleotide in the island P(CpG)/(expected)P(CpG)

FProm

Human promoter prediction

Method description:

Program predicts potential transcription start positions by linear discriminant function combining characteristics describing functional motifs and oligonucleotide composition of these sites. FProm uses file with selected factor binding sites from currently supported functional site data base.

For approximately 50-55% level of true promoter region recognition, FProm program will give one false positive prediction for about 4000 bp.

Another promoter recognition program, TSSG, uses promoter.dat file with selected factor binding sites (TFD, Ghosh,1993).

Prediction accuracy for each promoter type Promoter Type A: TATA-less promoter

Sensitivity	Specificity	Threshold*	Length**
1.000000	0.198215	-9.496	1.32975
0.990000	0.646996	-6.025	3.02029
0.950000	0.917724	-2.414	12.9585
0.900000	0.968909	+0.0467	34.2921
0.800000	0.992493	+3.329	142.028
0.700000	0.997591	+5.342	442.657

0.600000	0.998801	+6.508	889.255
0.500000	0.999409	+7.621	1805.3
0.400000	0.999705	+8.596	3610.59
0.300000	0.999858	+9.598	7491.98
0.200000	0.999911	+10.66	11987.2
0.100000	0.999968	+12.14	33297.7

Promoter Type B: TATA promoter

Sensitivity	Specificity	Threshold*	Length**
1.000000	0.773441	-6.766	71.1151
0.990000	0.965914	-2.318	472.68
0.950000	0.996183	+1.117	4220.83
0.900000	0.998333	+2.528	9667.06
0.800000	0.999570	+4.613	37459.9
0.700000	0.999785	+6.41	74919.8/td>
0.600000	0.999839	+7.963	99893
0.500000	0.999946	+9.586	299679
0.400000	0.999946	+11.21	299679
0.300000	0.999946	+12.5	299679
0.200000	1.000000	+14.14	1e+06
0.100000	1.000000	+16.54	1e+06

*Threshold value used by the program for a giver level of sensitivity

**Average length which contains 1 false-positive promoter.

References:

1. Solovyev V.V., Salamov A.A. (1997)
The Gene-Finder computer tools for analysis of human and model organisms genome sequences. In Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology (eds.Rawling C.,Clark D., Altman R.,Hunter L.,Lengauer T.,Wodak S.), Halkidiki, Greece, AAAI Press,294-302.

2. Solovyev V.V. (2001)
Statistical approaches in Eukaryotic gene prediction.
In Handbook of Statistical genetics (eds. Balding D. et al.), John Wiley & Sons, Ltd., p. 83-127.

3. Solovyev VV, Shahmuradov IA. (2003)
PromH: Promoters identification using orthologous genomic sequences. Nucleic Acids Res. 31(13):3540-3545.

FProm output:

FProm output:

```
Sequence     1 of     1,  Name: Homo sapiens chromosome 21; range 31946321 - 31958321;
length 12001
Length of sequence:     12001
     7 promoter/enhancer(s) are predicted
Promoter Pos:      6473 LDF:   +8.734
Promoter Pos:      3102 LDF:   +5.824
Promoter  Pos:       6078  LDF:   +16.297  TATA  box  at         6049     +5.597  TATAAAGT
Enchancer at:      5942 Score:  +12.499
Promoter Pos:      1363 LDF:   +5.235 TATA box at       1336    +6.514 AATAAAAG
Promoter Pos:      7068 LDF:   +1.165 TATA box at       7039    +4.190 TAAAAATA
Promoter Pos:      9650 LDF:   +1.051 TATA box at       9618    +4.491 GTTAAAAA
Promoter Pos:      5541 LDF:   +0.455 TATA box at       5512    +7.353 TATAAAAA
```

Where:

7 promoter/enhancer(s) are predicted	Number of predicted promoters in this sequence.
Each line below defines an appropriate predicted promoter. Detailed description of a line from this list is shown further: `6078 LDF:  +16.297 TATA box at       6049    +5.597 TATAAAGT Enchancer at: 5942 Score:  +12.499`	
Promoter Pos: 6078	Position of TSS on DNA.
LDF: +16.297	value of Fisher's linear discriminant for the current promoter. A bigger value corresponds to more reliable promoter.
If a promoter belongs to class of TATA-containing promoters, the following fields are added:	
TATA box at 6049	TATA-box position in the current promoter
+5.597	Score of this TATA-box
TATAAAGT	Nucleotide sequence of this TATA-box
If there is an enhancer in proximity to the current promoter, the following fields are added:	
Enchancer at: 5942	The position of enhancer in this promoter
Score: +12.499	Score of this enchancer

Parameters:

Input	
Sequence	Input file with sequence in FASTA-format
Output	
Result	Name of the output file
Print programm info	Print information about program accuracy. First and second type errors for each threshold value for each promoter type.

Nsite

Search for of consensus patterns with statistical estimation.

Nsite can be used for analysis of regulatory regions and composition of their functional motifs.

Method description:

The method is based on statistical estimation of expected number of a nucleotide consensus pattern in a given sequence [1-2,4]. It uses the Nsite formatted datafile, which can include any set of consensus sequences of functional motifs. In current version this file consists of the release of Transfac sequences (3.4, 1998, academic release), composite elements [3] and a set additional functional motifs.

If we find a pattern which has expected number significantly less than 1, it can be supposed that the analyzed sequence possesses the pattern's function.

In the output of Nsite we can see a pattern, its position in the sequence, accession number, ID, Description of motif and binding factor name from the original database if exist.

Table 1. Summary of single-letter code recommendations

Symbol	Meaning	Origin of designation
G	G	Guanine
A	A	Adenine
T	T	Thymine
C	C	Cytosine
R	G or A	puRine
Y	T or C	pYrimidine

M	A or C	aMino
K	G or T	Keto
S	G or C	Strong interaction (3 H bonds)
W	A or T	Weak interaction (2 H bonds)
H	A or C or T	not-G, H follows G in the alphabet
B	G or T or C	not-A, B follows A
V	G or C or A	not-T (not-U), V follows U
D	G or A or T	not-C, D follows C
N	G or A or T or C	aNy

Output example:

```
Program   NSITE  (Softberry Inc.)    | Version 2.2004
Search for motifs of   1500 Regulatory Elements (REs)     | SET of REs:
REGSITE DB (Transcription Regulatory Sites from human and animals) [ Last
Update: March 10, 2006]
____________________________________________________________
 Search PARAMETRS:
     Expected  Mean  Number                      :  0.0000000
     Statistical Siginicance Level               :  0.0000000
     Level of homology between known RE and motif:   80%
     Variation of Distance between RE Blocks     :   20%
 NOTE: RE - Regulatory Element/Consensus   | AC - Accession No of RE in a
given DB
       OS - Organism/Species   | BF - Binding Factor or One of them
       Mism. - Mismatches   | Mean. Exp. Number - Mean Expected Number   |
Up.Conf.Int. - Upper Confidence Interval
============================================================
 QUERY: >test_nsite.seq
 Length of Query Sequence:       2319 bp     | Nucleotide Frequencies:  A -
0.33   G -  0.19   T -  0.30   C -  0.18

............................................................
 RE:   620. AC: RSA00620//OS: chicken /GENE: BGP/RE: G-string /BF:
erythrocyte-specific protein
 Motifs on "-" Strand: Mean Exp. Number   0.00000     Up.Conf.Int.  1
Found   5
    2216  cGGGGGGGGGGGGGGG     2201 (Mism.= 1)
    2215  GGGGGGGGGGGGGGGG     2200 (Mism.= 0)
    2214  GGGGGGGGGGGGGGGG     2199 (Mism.= 0)
    2213  GGGGGGGGGGGGGGGG     2198 (Mism.= 0)
    2212  GGGGGGGGGGGGGGGt     2197 (Mism.= 1)
............................................................
 Totally       5 motifs of     1 different REs have been found
------------------------------------------------------------
```

Reference:

[1] Shahmuradov K.A. Kolchanov N.A.Solovyev V.V.Ratner V.A.
Enhancer-like structures in middle repetitive sequences of the eukaryotic genomes.
Genetics (Russ),22, 357-368,(1986).

[2] Solovyev V.V., Kolchanov N.A. 1994,
Search for functional sites using consensus In Computer analysis of Genetic macromolecules.
(eds. Kolchanov N.A., Lim H.A.),
World Scientific, p.16-21.

[3] Heinemeyer, T., Chen, X., Karas, H., Kel, A. E., Kel, O. V., Liebich, I., Meinhardt, T., Reuter, I., Schacherer, F., Wingender, E. (1999).
Expanding the TRANSFAC database towards an expert system of regulatory olecular

Solovyev V.V. (2002) Structure, Properties and Computer Identification of Eukaryotic genes. In Bioinformatics from Genomes to Drugs. V.1. Basic Technologies. (ed. Lengauer T.), p. 59 - 111.

Parameters:

Input	
Sequence	Name of the input file
Output	
Result	Name of the output file
Options	
DataBase	Select one of the site bases: **REGSITE DB (Animals)** **REGSITE DB (Plants)** **Animal TFD from Ghosh DB**
Mean Expected Number	Mean Expected Number
Minimal level of homology	Minimal level of homology
Statistical Significance Level	Statistical Significance Level
To allow variation	To allow variation
Data File with Right Boundaries positions	Data File with Right Boundaries positions

Nsite-h

Search for functional motifs conserved in orthologs

ACTION:

Search for Conservative Motifs of Regulatory Elements (REs)from both Collection of thousands REs (of human and animals or plant species) created by us and Collection of REs given by USER available in both of 2 aligned (in special FORMAT) homologous (orthologous) DNA sequences (Max. Length - 100 000 nt)

SEARCH CONDITIONS:

(1) Expected Mean Numbers of any regulatory motif found must be less than a given number (default: 0.01);

(2) Homology Level of any motif in one sequence with the corresponding area of another sequence (in relation to ALIGNMENT) must be higher than a given level.

Output example:

```
Program   Nsite-h  (Softberry Inc.)    | Version 2.2004
Search for motifs of    702 Regulatory Elements (REs) in a pair of Homologous
Sequences

    | SET of REs: REGSITE DB (Plants; version IV)
_____________________________________________________________
 Search PARAMETERS:
     Expected  Mean  Number                 :  0.0500000
     Statistical Significance Level          :  0.9500000
     Minimal Conservative Level             :  80 %
     Level of homology between known RE and motif:   80%
     Variation of Distance between RE Blocks     :   20%
 NOTE: RE - Regulatory Element/Consensus    | AC - Accession No of RE in a
given DB
       OS - Organism/Species   | BF - Binding Factor or One of them
       Mism. - Mismatches   | Mean. Exp. Number - Mean Expected Number
```

```
         | Up.Conf.Int. - Upper Confidence Interval
============================================================
            QUERY:            >H-NPPA/AL021155/[33199:35843/c]/-2000:+645/CDS:
33198/c,premRNA:>33843/c
 Length of Query Sequence:       2845 bp

    | Nucleotide Frequencies:  A -  0.25   G -  0.27   T -  0.24   C -  0.24

............................................................
 RE:       1. AC: RSP00001//OS: Spinach /GENE: rps1/RE: S1F_BS /BF: S1F,
spinach leaf nuclear factor
 Motifs on "+" Strand: Mean Exp. Number    0.00090       Up.Conf.Int.   1
Found   1
    2577  AGAATTGTTACCATGAAA     2594 (Mism.= 0; Cons.: 100 %)
............................................................
 RE:      2. AC: RSP00002//OS: Brassica napus /GENE: Oleosin/RE: ABRE-3 /BF:
B.napus embryo protein factor
 Motifs on "+" Strand: Mean Exp. Number    0.01145       Up.Conf.Int.   1
Found   1
    2619  ACACGTGGC     2627 (Mism.= 0; Cons.: 100 %)
............................................................
 RE:      4. AC: RSP00004//OS: Arabidopsis thaliana /GENE: CHS/RE: UV/BLRE
/BF:unknown
 Motifs on "+" Strand: Mean Exp. Number    0.03635       Up.Conf.Int.   1
Found   1
    2628  TAGACACGTAGA     2639 (Mism.= 0; Cons.: 100 %)
............................................................
 RE:       6. AC: RSP00006//OS: Soybean, Glysine max /GENE: GS15/RE: ATRE
/BF:unknown
 Motifs on "+" Strand: Mean Exp. Number    0.00728       Up.Conf.Int.   1
Found   1
    2651  AAATTATTTTATAT     2664 (Mism.= 0; Cons.: 100 %)

 Motifs on "-" Strand: Mean Exp. Number    0.00763       Up.Conf.Int.   1
Found   1
     831  AAATgATTTTATtT      818 (Mism.= 2; Cons.: 100 %)
............................................................
 RE:      7. AC: RSP00007//OS: Tobacco; Nicotiana tabacum /GENE: CHN50/RE:
ElRE /BF: unknown
 Motifs on "+" Strand: Mean Exp. Number    0.00003       Up.Conf.Int.   1
Found   1
    2665  GATTTGGTCAGAAAGTCAGTCC     2686 (Mism.= 0; Cons.: 100 %)
............................................................
 RE:      8. AC: RSP00008//OS: Spinach; Spinachia oleracera /GENE: NiR/RE:
NiRE /BF: NIT2 ZN-finger protein
 Motifs on "+" Strand: Mean Exp. Number    0.00000       Up.Conf.Int.   1
Found   1
    2687  CAAAGCGACAAAAATAGATATTAGTAACACA     2717 (Mism.= 0; Cons.: 100 %)
............................................................
 RE:      9. AC: RSP00009//OS: Spinach; Spinachia oleracera /GENE: NiR/RE:
GATA /BF: NIT2 ZN-finger protein
 Motifs on "+" Strand: Mean Exp. Number    0.02504       Up.Conf.Int.   1
Found   3
     2466  TAGATA      2471  --24--      2496  TATCTA      2501 (Mism.= 0/ 0;
Cons.: 100/100 %)
     2502  TAGATA      2507  --25--      2533  TATCTA      2538 (Mism.= 0/ 0;
Cons.: 100/100 %)
     2539  TAGATA      2544  --26--      2571  TATCTA      2576 (Mism.= 0/ 0;
Cons.: 100/100 %)

 Motifs on "-" Strand: Mean Exp. Number    0.02573       Up.Conf.Int.   1
Found   3
     2576  TAGATA      2571  --26--      2544  TATCTA      2539 (Mism.= 0/ 0;
Cons.: 100/100 %)
```

```
    2538   TAGATA      2533  --25--      2507   TATCTA      2502 (Mism.= 0/ 0;
Cons.: 100/100 %)
    2501   TAGATA      2496  --24--      2471   TATCTA      2466 (Mism.= 0/ 0;
Cons.: 100/100 %)
............................................................
  RE:      11.  AC:  RSP00011//OS:  Catharanthus  roseus  /GENE:  Str/RE:  G-box
(ext) /BF: TAF-1
 Motifs  on  "+"  Strand:  Mean  Exp.  Number    0.01262       Up.Conf.Int.   1
Found   1
   2778  CTCCACGTGGT     2778 (Mism.= 0; Cons.: 100 %)
............................................................
...
```

Parameters:

Input	
Sequences 1	Name of the 1-st input file
Sequence 2	Name of the 2-nd input file
Output	
Result	Name of the output file
Options	
DataBase	Select one of the site bases: **REGSITE DB (Animals)** **REGSITE DB (Plants)** **Animal TFD from Ghosh DB**
Conservative Level	Conservative Level
Mean expected number	Mean expected number.
Statistical siginicance level	Statistical siginicance level.
Minimal level of homology	Minimal level of homology between Known RE/consensus and motif found.

Nsite-m

Search for regulatory motifs conserved in several sequences.

Regulatory Elements (REs) can be taken from different databases or defined by user (for local runs only). The program finds sites that occur at least in one copy in P% or more of analyzed DNA sequences (in web version P is set to 50%). Input sequences should be in FASTA format, like

```
>test1
AAAAAAAAA
GGCCCCCCC
>test2
ACCCTTTTTC
CCCCCCCCCC
```

Method description

As Nsite, Nsite-m is also based on search of statistically significant regulatory site consensus - see NSITE Help for more description.

The main features of the approach are the follows:

(i) RE may consist of a single box (a continuous DNA segment) or two boxes, spaced by some DNA sequence, where only length, but not nucleotide content, of this spacer is important for functioning of such a composite site.

(ii) A real RE or its IUPAC consensus contains both variable positions, where the presence of a certain group of nucleotides is permissible, and strictly conserved positions, where strict identity between real site/consensus and predicted motif is required . The nonequivalence of these

positions should be taken into account, i.e., complete homology at conserved positions is required, and a violation of homology in the variable positions should be permissible.
(iii) The homology between RE and a motif on query DNA sequence may be a random happening, therefore, estimation of its statistical significance is very important. A conclusion on functional significance of revealed homology can be reached only if the homology is significantly nonrandom, i.e., the homology is not a random event.
(iv) Characteristics such as nucleotide frequencies should not be used when describing consensus because of its small size. Instead, one should use estimates based on number of specific nucleotides in the consensus.
(v) Although all available RE databases usually annotate fixed distance between two boxes of composite elements, some variability of the spacer length usually takes place. Therefore, search algorithm for composite REs should allow some limited flexibility in spacer length.

Expected occurency for each regulatory motif found must be less than given percentage (default: 5%);
The program currently uses Transfac human/animal and plant datasets (3587 and ~600 real sites/consensuses, respectively). User can perform a search for motifs of REs from his own dataset in a format described below.

Nsite-m output

Output file begins with description of the program allocation, search parameters, as well as, if using our datasets, abbreviations used. Two next lines include name and length of the first query sequence. Then, statistical analysis of search result are presented. At last, names of REs, statistical estimation and sequences of motifs found and are given.

```
 Program   Nsite-m: Search for Motif Patterns (Softberry Inc.)
____________________________________________________________
 File with QUERY Sequences: H-H.SEQ
 Search PARAMETERS:
     Expected  Mean  Number                 :  0.0100000
     Print  Query  Sequence                 : No
     Special numbering of Query Sequence    : No
     Variation of Distance between RE Blocks: No
     Create List of Numbered Query Sequences: No

 NOTE: RE - Regulatory Element/Consensus
       AC - Accession No of RE in TRANSFAC
       OS - Organism/Species
       BF - Binding Factor or One of them
       Mism.            - Mismatches
       Mean. Exp. Number - Mean Expected Number
============================================================
 STATISTICAL ANALYSIS of RESULTS of SEARCH of MOTIFS
       of 3587 REs in    5 SEQUENCES
============================================================
 Motif(s) of  2 REs in  50 %  or more of analyzed sequences


 RE:   429. AC: R00560  OS: human  BF: CACCC-binding
   ctccacccatggg
 RE:  1272. AC: R01859  OS: human  BF: CP1
   gccttgaccaat

 FOUND in every of the following    3 ( 60.00 % of all) sequences:
     3    4    5
............................................................
 RE:   738. AC: R01053  OS: mouse  BF: RXR-beta
   tgaggtcaggg
 RE:  2751. AC: R03786  OS: empty  BF: PUB1
   tttatttatgttttcttctgca
 FOUND in every of the following    3 ( 60.00 % of all) sequences:
     1    4    5
________________________________________________________________
SUMMARY: In 2 case(s)  motif(s) of  2 REs found in  50 % or more of analyzed
sequences

=====================================================
     Motifs of REs found in  50 %  or more of analyzed sequences
................................................................

   1. QUERY: >GB/U01317.1|Human HBB (H-HBB) [60137-->2500 nt]: -2000...+500

 Length of Query Sequence:       2150
 Nucleotide Frequencies:  A -  0.32   G -  0.20   T -  0.30   C -  0.17

................................................................
 RE:   738. AC: R01053  OS: mouse  BF: RXR-beta
           (Found in    3 ( 60.00 %) SEQs)

 Motifs on "-" Strand: Mean Exp. Number   0.00459    Found  1

     783  TGAGGTCAGcG      773 (Mism.= 1)
==================================================================================
=
```

RULES for creating USER RE sets:

1. User sets must include only sequences of actual REs and/or their consensus sequences.

2. Every actual RE/consensus is described in three lines:
   LINE 1: Name/description of RE/consensus
   LINE 2: Sequence of of RE/consensus
   LINE 3: <par1> <par2> <par3> <par4>

3. Sequence (LINE2) may include both standard nucleotides (A/a, T/t, G/g,C/c)
and their combinations according to IUPAC abbreviations:
R - A or G, Y - T or C, K - G or T, M - A or C, S - G or C,
W - A or T, B - G or T or C, D - A or G or T, H - A or C or T,
V - A or G or C, N - A or G or C or T.

In the case of composite REs, two boxes are seperated by "-".

Length of RE/consensus sequence must not exceed 80 symbols, including "-" in case of composite elements.

Capital letters indicate Conservative nucleotides (positions) in which mismatch is not allowed.

4. In the LINE 3: <par1> - maximal number of mismatches for the first box
<par2> - maximal number of mismatches for the second box (for composite REs).
If RE contains a single box, then <par2> = 0;
If any mismatch is not allowed, then <par1> = <par2> = 0.

<par3> - minimal distance between boxes of composite RE

```
                 <par4> - maximal distance between boxes of composite RE
                          (for a single-box REs <par3> = <par4> = 0 )

 All <par1> <par2> <par3> and <par4> are given as INTEGERS in 4i5 format.


 Example of USER's set of 3 REs:

 RE 1
 agTGGcgAggcg
     2    0    0    0
 RE2
 caggccTGc-CCAGctgg
     1    1    8   10
 RE 3
 RRTGTGGWWW
     0    0    0    0
```

Parameters:

Input	
Sequences	Name of the input file
Output	
Result	Name of the output file
Options	
DataBase	Select one of the site bases: **REGSITE DB (Animals)** **REGSITE DB (Plants)** **Animal TFD from Ghosh DB**
Mean Expected Number	Mean Expected Number
Minimal level of homology	Minimal level of homology
Statistical Significance Level	Statistical Significance Level
To allow variation	To allow variation
Data File with Right Boundaries positions	Data File with Right Boundaries positions

Pattern

Search for significant patterns in the set of sequences.

Pattern output:

Example of output:

```
Total sequences: 20
Found 10 pattern(s)
Pattern      1, Length:    9, Power:    20(100%), Q:70.699721, Inf:11.5212
( 2.3555) Q2:70.699721, F0:    2.24981
Consensus: CGCABHBGG
Initial:   GCTATCGG
Frequences:
   A    C    G    T
   0  950   50    0   1.7136
   0  100  850   50   1.2524
   0  950   50    0   1.7136
 850    0   50  100   1.2524
 200    0    0  800   1.2781
  50    0  200  750   1.0082
 200  700   50   50   0.7432
 150   50  750   50   0.8460
   0   50  950    0   1.7136
Sequences:
     1:    126    134 + CGCATTCGG *      6636
```

```
      2:    186    194 + CGCTATAGG *      4047
      3:    239    247 + CGCATTCGC *      5341
      4:    212    220 + CGCATGCAG *      5029
      5:    251    259 + CGCATGCGG *      5888
      6:    456    464 + CGCATGGGG *      4804
      7:    183    191 + CGGATTCTG *      4203
      8:    103    111 + CCCGTTCGG *      4342
      9:    492    500 + CTCATTCCG *      4302
     10:    468    476 + CGCATTCGG *      6636
     11:    509    517 + CGCAATCGG *      5845
     12:    495    503 + CGCAATCGG *      5845
     13:    219    227 + GCCATTCGG *      4254
     14:    434    442 + CGCATTTGG *      5551
     15:    280    288 + CGCATGCGG *      5888
     16:    430    438 + CGCTATCGG *      4759
     17:    337    345 + CGCATTAGG *      5924
     18:     99    107 + CGCATAAGG *      4810
     19:    133    141 + CGCATTCAG *      5777
     20:    521    529 + CGCATTAAG *      5065

Pattern       2,  Length:     9,  Power:     19(95%),  Q:66.807998,  Inf:11.7074
( 2.3381) Q2:66.807998, F0:    2.16649
Consensus: CGCATTCGG
Initial:   GCATTCAG
Frequences:
   A    C    G    T
   0  947   53    0   1.7025
   0  105  842   53   1.2258
   0  947   53    0   1.7025
 895    0   53   53   1.4093
 158    0    0  842   1.3708
  53    0  211  737   0.9785
 158  737   53   53   0.8077
 158   53  737   53   0.8077
   0   53  947    0   1.7025
Sequences:
      1:    126    134 + CGCATTCGG *      6642
      3:    239    247 + CGCATTCGC *      5374
      4:    212    220 + CGCATGCAG *      5117
      5:    251    259 + CGCATGCGG *      5935
      6:    456    464 + CGCATGGGG *      4838
      7:    183    191 + CGGATTCTG *      4271
      8:    103    111 + CCCGTTCGG *      4367
      9:    492    500 + CTCATTCCG *      4375
     10:    468    476 + CGCATTCGG *      6642
     11:    509    517 + CGCAATCGG *      5732
     12:    495    503 + CGCAATCGG *      5732
     13:    219    227 + GCCATTCGG *      4320
     14:    434    442 + CGCATTTGG *      5544
     15:    280    288 + CGCATGCGG *      5935
     16:    430    438 + CGCTATCGG *      4494
     17:    337    345 + CGCATTAGG *      5813
     18:     99    107 + CGCATAAGG *      4734
     19:    133    141 + CGCATTCAG *      5824
     20:    521    529 + CGCATTAAG *      4995
...
```

Where

Total sequences: 20	- number of sequences that formed a pattern.
Found 10 pattern(s)	- number of patterns.
Pattern 1	- pattern's number.
Length: 9	- length of pattern's sequences.
Power: 20(100%)	- number and percentage of sequences that were included into

	pattern.
Q:70.699721	- quality of a pattern that reflects both its homogeneity and its power.
Inf:11.5212 (2.3555)	- informational content of a pattern.
Q2:70.699721	- quality of a pattern in the context of its presentation's skew in target and control sets.
F0: 2.24981	- indicates the frequency of occurrence in a target set.
Consensus: CGCABHBGG	- consensus of a pattern for 15-letter alphabet.
Initial: GCTATCGG	- initial consensus, from which the pattern was created.
Frequences:	- pattern's matrix of frequencies. The right column represents an informational content of each pattern's position:
Sequences:	- weight of all sequences that formed a pattern.
1: 126 - 134	- start and end of sequences that formed a pattern.
+	- strand direction.
CGCATTCGG *	- sequence of a pattern. * means that this sequence was used in pattern formation.
6636	- weight of a pattern in matrix of frequencies.

Parameters:

Input	
Sequence	Input file - nucleotide sequences in FASTA-format
Output	
Result	Name of the output file
Print N best patterns pairs	Print N best patterns pairs
Options	
Search in both chain	Search for pattern in both chain
Threshold for include fragment	Threshold for include fragment to pattern.
Minimal distance for patterns in pair	Minimal distance for patterns in pair
Maximal distance for patterns in pair	Maximal distance for patterns in pair
Number of stored best patterns	Number of stored best patterns
Initial length	Initial length. Minimal value is 3, maximal value is 12.
Try to expand	Try to expand to xx position left and right. If this option is switched off, the pattern will not extend in the parties. Default value is 2, minimal value is 1, maximal value is 10.
Pair selection methods	Pair selection methods: **Both pattern must present** **One of pattern must present**

PolyaH

Recognition of 3'-end cleavage and polyadenilation region of human mRNA precursors

Method description:

Algorithm predicts potential position of poly-A region by linear discriminant functions combining characteristics describing various contextual features of these sites. The default LDF threshold in the server is equal 0.

Accuracy:

The accuracy has been estimated for the set of 131 poly-A regions and 1466 non-poly-A regions of human genes, having AATAAA sequence. For 86% accuracy poly-A region prediction the algorithm has 8% false predictions (Sp=50%; C=0.62). For example, with threshold 0.7 it predicts 8 of 9 poly-A sites of AD2 genome (35937 bp.) and overpredict 4 false (Compare with method of poly-A site prediction (CABIOS 1994,10,597-603), which for 8 true predicted sites gives 968 false positive sites).

PolyaH output:

First line - name of your sequence; 2nd line - Length of your sequence

Next lines - positions of predicted sites and their 'weights', Position shows the first nucleotide of the AATAAA consensus in the predicted region

For example:

```
 HSG11C4A     1741 bp    DNA             PRI       21-FEB
 Length of sequence-      1741
     1 potential polyA site was predicted
 Pos.:    988 LDF-  4.06
```

Parameters:

Input	
Sequence	Name of the input file
Output	
Result	Name of the output file

PromH-AN

Search for animal promoters using 2 homologous 5'-regions.

Method description

To further improve promoter identification accuracy achieved by TSSG program, we developed a new program, promH(G), by extending the TSSG program feature set. PromH uses linear discriminant functions that take into account, in addition to features realized in TSSG, conservation features of major promoter functional components, such as transcription start points, TATA-boxes and regulatory motifs, in pairs of orthologous genes aligned by SeqMatch-N program.

PromH(G) output

OUTPUT file begins with description of the program allocation, used abbreviations and Search Parameters (Lines 1-10). Next two lines include name and length of the first query sequence and the number of predicted promoter regions. Then, positions of predicted sites, their "weights" and TATA-box position (for TATA promoters) are given. After that, functional motifs are given for every predicted region; (+) and (-) reflect direct or complementary chain; $... means a particular motif identificator from Transcription Factors Database, TFD (Ghosh, Nucleic Acids Res., 1993 , 21, 3117-3118). Then, the same information is given for second query sequence.

```
Program  promHG  (Softberry Inc.)
      Search for TATA+/TATA- promoters in 2 aligned DNA sequences

  NOTE: PHa  - Homology Level of Aligned Sequences in LOCAL Search Area (-
100,TSS+40)
        PHs  - Homology Level of Aligned Sequences around TSS
        PHss - Homology Level of Aligned Sequences to Right from TSS
        PHt  - Homology Level of TATA-boxes in Aligned Sequences
         PHr  - Mean Homology Level of Regulatory Elements in LOCAL Search
Area

    Initial / Final Thresholds -   2.00 /  6.00
 ==================================================================
 >H-NPPA/AL021155/[33199:35843/
  Length of sequence-      2645
```

```
       1 promoter(s) have been predicted
  Promoter Pos:   2549 (Weight - 16.00)   TATA box at:   2517 (Weight -
218.33)
     PHa -  78%   PHs - 100%   PHss -  74%   PHt - 100%   PHr -  80%
  Transcription factor binding sites:
 for promoter at position -   2549
   2462 (+) S01152      AAGTGA
   2378 (+) S00922      AGAGG
   2525 (+) S00922      AGAGG
   2306 (-) S00922      AGAGG
   2499 (-) S00395      CACGCW
 ..............


 ----------------------------------------------------
 >R-NPPA/J03267/[1638:3722]/-2000:+85/CDS: 3723, premRNA: 3638
 Length of sequence-     2087
       2 promoter(s) have been predicted
  Promoter Pos:   2000 (Weight - 15.59)   TATA box at:   1970 (Weight -
217.73)
     PHa -  78%   PHs - 100%   PHss -  77%   PHt - 100%   PHr -  89%
  Promoter Pos:   1662 (Weight:   6.37)
     PHa -  76%   PHs -  88%   PHss -  72%   PHr -  74%
  Transcription factor binding sites:
 for promoter at position -   2000
   1915 (+) S01152      AAGTGA
   1773 (-) S00922      AGAGG
   1716 (+) S00392      AGGAAG
   1999 (-) S02113      CCAGCTG
   1713 (+) S01003      CCCAG
 ...........
 for promoter at position -   1662
   1504 (+) S01090      AATGA
   1610 (+) S01013      ACAGCTG
   1484 (+) S00922      AGAGG
   1505 (+) S01444      ATGAATCAG
 ...........
```

Parameters:

Input	
Sequence 1	Name of the input file
Sequence 2	Name of the input file
Ouput	
Result	Name of the output file

ScanWM-PL

The program for site search in DNA sequences by score matrices.

The program's brief description.

ScanWM-PL is a program that search for motifs in "+" and "-" strands of DNA using score matrices. The program takes DNA sequences one by one from FASTA file, takes matrices from the score matrices file and annotates DNA sequences by finding motifs (potential sites for binding of transcription factors) in accordance to score matrices. Nucleotide sequences are referred to as motifs (potential sites for binding of transcription factors) if their score is more or equal to "cut-off value" of score matrix; at that the score of sequence is calculated as sum of its nucleotides' score, and the score of a nucleotide in appropriate position is defined in accordance to score matrix. Since ScanWM works with score matrices, elements of which are "log likelihood ratios", the summation is used at sequence score detection.

Algorithm.
In the current version of the program there is no checking for overlapping motifs. Checking for overlapping motifs could be of importance for motifs of those sites, sequences of which can be read similarly (or almost similarly) in both forward and backward orientations.
Definition of the data volumes.

Initially, the program does not know the approximate number of motifs, that can be found in a single sequence using a single score matrix.
For storing motifs the dynamic container is used. If, at a certain step, the number of motifs becomes greater than the current volume of container, then its volume increases by the number of elements, defined by the "increment"-value of the container's volume.
In the current version of the program, the initial and "increment-" volumes of container for motifs are set equal to 100 and 100.
FASTA file.

In the current version of program, the maximal number of symbols in a line of FASTA file = 999.

Format of a file with score matrices

Score matrices in a score matrices file have the following record format:

```
2. AC: RSP00002//OS: Brassica napus /GENE: Oleosin/RE: ABRE-3 /BF: ...

      1430      9.29   10.28   12.76    6.79    1.49

        1       2       3       4       5       6       7       8       9
A     0.96   -2.46    1.12   -2.57   -2.76   -3.49   -3.24   -2.12   -1.15
C    -0.44    1.63   -4.85    1.65   -3.60   -3.47   -3.47   -2.12    1.53
G    -2.55   -2.02   -3.47   -2.72    1.67  -10.16    1.69    1.38   -1.91
T    -2.34   -2.36   -3.29   -2.66   -2.91    1.12   -3.49   -0.37   -2.06
```

Each score matrix takes 10 lines in a file.
The first line - ID-line of a score matrix;
The third line - "line of values" (see below);
The fifth line - score matrix's positions;
The sixth to ninth lines - the score matrix itself (in a format, shown above).
The empty lines: second, fourth and tenth ones.
Format and table-description of "values' lines".

```
 1430      9.29   10.28   12.76    6.79    1.49
```

value (example)	Description
1430	Number of sequences, used to build the score matrix.
9.29	Site's IC
10.28	Average score (*)
12.76	Maximal score (*)
6.79	Minimal score (*)
1.49	Standard deviation (*)

(*) Using the matrix, the scores for sequences, used to build the matrix, are calculated, and average, maximal and minimal scores as well as standard deviation are revealed.
In the current version of ScanWM, if -t: parameter is set to 1, i.e. -t:1, then of all "values' line" numbers the average score and standard deviation (see table) only are used. Other "values' line" numbers are not used, and at preparation of user-defined files with score matrices can be set, for example, to zero.

Format of a file with results of searching for motifs using score matrices

Format of a file with results of searching for motifs using score matrices has a following structure.
In the header, the data on a program version and parameters used for program launch are shown:

```
Program  ScanWM  (Softberry Inc.)

 Search for motifs by Weight Matrixes of Regulatory Elements
 Version 1.2004

 SET of WMs: derived from subsection of REGSITE DB (Plants; version IV)
____________________________________________________________

 File with QUERY Sequences: TEST_SEQ.seq

 Search PARAMETERS:
     Threshold type                       : 2
     Threshold value                      : 0.90
     Search for motifs on "+" strand      : yes
     Search for motifs on "-" strand      : yes

 NOTE: WM - Weight Matrix of Regulatory Element
       AC - Accession No  of Regulatory Element in a given DB
       OS - Organism/Species
       BF - Binding Factors or One of them

============================================================
```

Further, for each DNA sequence (from designated set), there are located its ID-string and length followed by results of searching for motifs using score matrices: for each of the score matrices, the ID-string and motifs found on "+" and/or "-" strands of DNA are shown;
For each of found motifs, there are shown its sequence, coordinates in "QUERY sequence" and a score, obtained using a score matrix;
Motifs, found on "-" strand, are shown in 5'-3' orientation, and thus, since coordinates are shown relatively to "+" strand (which corresponds to "QUERY sequence"), the first coordinate should be greater then the second one (see example below);
In the end, the total number of motifs, found in a sequence, and the total number of score matrices, used for search, are shown.
Below there is an example of output for a single sequence and a single score matrix (ID-string of a sequence and ID-string of a score matrix are shown incompletely):

```
------------------------------------------------------------

 QUERY: >At4g00860 stress-related ozone-induced protein (OZI1)...
 Length of Query Sequence:        350

............................................................
 WM:     228. AC: RSP00231//OS: Arabidopsis thaliana /GENE: AGAMAOUS (AG)...

 Motifs on "+" strand (in DIR orientation):  Found   1

     121  CCAATCT      127       7.73
 Motifs on "-" strand (in INV orientation):  Found   1

     192  CCCATCT      186       6.65

............................................................
 Totally       2 motifs  of     1 different WMs have been found
------------------------------------------------------------
```

If no motifs were found in a sequence, then output for this sequence is displayed as following:

```
------------------------------------------------------------

 QUERY: >At1g04660 68414.t00411 glycine-rich protein
 Length of Query Sequence:        350

............................................................
 Any Motif not found
------------------------------------------------------------
```

OUTPUT EXAMPLE

The whole output of ScanWM-PL for some test sequence is shown below.

```
Program  ScanWM  (Softberry Inc.)

 Search for motifs by Weight Matrixes of Regulatory Elements
 Version 1.2004

 SET of WMs: derived from subsection of REGSITE DB (Plants; version IV)
____________________________________________________________

 File with QUERY Sequences: TEST_SEQ.seq

 Search PARAMETERS:
     Threshold type                        : 2
     Threshold value                       : 0.90
     Search for motifs on "+" strand       : yes
     Search for motifs on "-" strand       : yes

 NOTE: WM - Weight Matrix of Regulatory Element
       AC - Accession No  of Regulatory Element in a given DB
       OS - Organism/Species
       BF - Binding Factors or One of them

============================================================

 QUERY: >At4g00160 [-300,+50] region of F-box family protein
 Length of Query Sequence:        350

............................................................
  WM:        >151.  AC:  RSP00151//OS:  tomato,  Lycopersicon  esculentum  /GENE:
Lhcb1*1, Lhcb1*2, Lhca3, Lhca4/RE: CRE, consensus /BF:unknown

 Motifs on "+" strand (in DIR orientation):  Found   1

      79  CAAGTACATC       88       7.76

............................................................
 WM:     >174. AC: RSP00174//OS: Phaseolus vulgaris /GENE: beta-phaseolin, or
phas/RE: ATCATC motif /BF:unknown

 Motifs on "+" strand (in DIR orientation):  Found   2
```

```
      21  ATCATC       26       7.98
     102  ATCATC      107       7.98

.............................................................
  WM:        >359.  AC:  RSP00359//OS:  barley,  Hordeum  vulgare  /GENE:  GCCGAC
motif/RE: HVA1s /BF: HvCBF1

 Motifs on "-" strand (in INV orientation):  Found   1

     103  ATCGAC       98       4.73

.............................................................
  WM:        >707.  AC:  RSP00707//OS:  /GENE:  /RE:  W-box  (consensus  1)  /BF:
transcription factors of WRKY family

 Motifs on "-" strand (in INV orientation):  Found   3

     120  AATGACC      114       4.56
     137  AATGACC      131       4.56
     286  AATGACT      280       4.42

.............................................................
 WM:     >722. AC: RSP00722//OS: Nicotiana plumbaginifolia /GENE: rbcS 8B/RE:
I-box /BF: unknown transcription factor

 Motifs on "-" strand (in INV orientation):  Found   1

     251  GATAAGA      245       9.12

.............................................................
 Totally       8 motifs  of     5 different WMs have been found
```

Parameters:

Input	
Sequences	File with fasta sequences. In the current version of program, the maximal number of symbols in a line of FASTA file = 999.
Output	
Result	Name of the output file
Options	
Threshold type	threshold type, formula to calculate weight matrix cut-off value: **Based on weights of training motifs** - formula is: *Cut-off = Average + THR_VALUE * Std_dev* *"Average"* and *"Std_dev"* (standard deviation) are calculated for weights of motifs from which a weight matrix has been built. *THR_VALUE* is a real number (including 0). *THR_VALUE* is specified by "Threshold value" option. **Based on similarity to weight matrix** - formula is: *Cut-off = WM_Min_Value + THR_VALUE * (WM_Max_Value - WM_Min_Value)* *"WM_Min_Value"* and *"WM_Max_Value"* are minimal and maximal values that can be obtained with a corresponding weight matrix. *THR_VALUE* must belong to interval [0;1] (with default value = 0.9). *THR_VALUE* is specified by "Threshold value" option.
Threshold value	threshold value
DNA chain	DNA chain: **Direct** **Reverse**

	Both

TSSG

Recognition of human PolII promoter region and start of transcription

TSSG is the most accurate mammalian promoter prediction program. The following table shows results of promoter search on genes with known mRNAs by different promoter finding programs, reproduced with changes from Liu and States (2002) Genome Research 12:462-469. It shows that TSSG has by far the fewest false positive predictions.

Parameters:

Program	**Set1 (133 promoters)**		**Set2 (120 promoters)**	
	True predictions	False Predictions	True predictions	False Predictions
PROSCAN1.7	32 (24%)	18 (36%)	30 (25%)	22 (42%)
NNPP2.0	56 (42%)	41 (42%)	26 (22%)	50 (66%)
PromFD1.0	88 (66%)	43 (33%)	69 (58%)	57 (45%)
Promoter2.0	8 (6%)	100 (93%)	14 (12%)	92 (88%)
TSSG	75 (56%)	10 (12%)	62 (52%)	18 (23%)
TSSW	57 (43%)	29 (34%)	58 (48%)	20 (26%)

Method description:

Algorithm predicts potential transcription start positions by linear discriminant function combining characteristics describing functional motifs and oligonucleotide composition of these sites. TSSG uses promoter.dat file with selected factor binding sites (TFD, Ghosh,1993) developed by Dan Prestridge to calculate the density of functional sites as in J.Mol.Biol.,1995,249,923-932.

For approximately 50-55% level of true promoter region recognition, TSSG program gives one false positive prediction for about 5000 bp. This accuracy is similar with the test sequences anlysis by Prestridge's method. We estimate an accuracy of finding TSS position on ten test genes where both our and Prestridge's algorithms found promoter region to be as follows (numbers show dictance between actual and predicted TSS):

Method/distance	**<5bp**	**5-50 bp**	**50-150 bp**	**Mean of observed distance**
Prestridge's	0	3	7	81.2 bp
TSSG	7	3	0	7.3 bp

Another Softberry promoter recognition program TSSW is based on similar ideology, but uses data from older release of Biobase's Transfac® data base (E.Wingender, J.Biotech., 1994, 35, 273-280).

References:

1. Solovyev V.V., Salamov A.A. (1997)
The Gene-Finder computer tools for analysis of human and model organisms genome sequences. In Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology (eds.Rawling C.,Clark D., Altman R.,Hunter L.,Lengauer T.,Wodak S.), Halkidiki, Greece, AAAI Press,294-302.
2. Solovyev V.V. (2001)
Statistical approaches in Eukaryotic gene prediction.
In Handbook of Statistical genetics (eds. Balding D. et al.), John Wiley & Sons, Ltd., p. 83-127.
3. Solovyev VV, Shahmuradov IA. (2003) PromH: Promoters identification using orthologous genomic sequences. Nucleic Acids Res. 31(13):3540-3545.

TSSG output:

First line - name of your sequence;
second and third lines - LDF threshold and the length of presented sequence
Fourth line - Number of predicted promoter regions
Next lines - positions of predicted sites, their 'weights' and TATA box position (if found)

Position shows the first nucleotide of the transcript (TSS position)
After that functional motifs are given for each predicted region; (+) or (-) reflects the direct or complementary chain; `Fields like "RSP00004      tagaCACGTaga"  mean a particular motif`
`>identificator with found similar sequence from the Softtberry`
`>Regsite-Plant data base.`

For example:

```
 HSCALCAC     7637 bp    DNA             PRI       14-MAR-1995
 Length of sequence-      7637
 Threshold for LDF-  4.00
     1 promoter(s)  were predicted
 Pos.:   1820 LDF- 16.65 TATA box predicted at   1804
 Transcription factor binding sites:
for promoter at position -    1820
  1764 (-) S00098       AACCAAT
  1608 (-) S01152       AAGTGA
  1741 (+) S01153       AARKGA
  1608 (-) S01153       AARKGA
  1657 (+) S01090       AATGA
  1617 (-) S01027       ACGCCC
  1577 (+) S00534       ACGTCA
  1580 (-) S00534       ACGTCA
  1580 (-) S01257       ACGTCAT
..............................
```

Lower cased letters mean non-conserved nucleotides in the site consensus

The letters except (A,T,G,C) describe ambiguous sites in a given DNA sequence motif, where a single character may represent more than one nucleotide using Standard IUPAC Nucleotide code.

See TABLE at http://www.yeastract.com/help/help_searchbydnamotif.php#Ref1

IUPAC Code	**Meaning**	**Origin of Description**
G	G	Guanine
A	A	Adenine
T	T	Thymine
C	C	Cytosine
R	G or A	puRine
Y	T or C	pYrimidine
M	A or C	aMino
K	G or T	Ketone
S	G or C	Strong interaction
W	A or T	Weak interaction
H	A or C or T	not-G, H follows G in the alphabet
B	G or T or C	not-A, B follows A in the alphabet
V	G or C or A	not-T (not-U), V follows U in the alphabet
D	G or A or T	not-C, D follows C in the alphabet

N	G or A or T or C	aNy

Parameters:

Input	
Sequence	Name of the input file
Output	
Result	Name of the output file

TSSP

Recognition of human Pol II promoter region and start of transcription

Method description:

Algorithm predicts potential transcription start positions by linear discriminant function combining characteristics describing functional motifs and oligonucleotide composition of these sites. TSSP uses file with selected factor binding sites from RegSite DB (Plants) developed by Softberry Inc.

References:

1. Solovyev V.V., Salamov A.A. (1997)
The Gene-Finder computer tools for analysis of human and model organisms genome sequences.
In Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology (eds.Rawling C.,Clark D., Altman R.,Hunter L.,Lengauer T.,Wodak S.), Halkidiki, Greece, AAAI Press,294-302.
2. Solovyev V.V. (2001)
Statistical approaches in Eukaryotic gene prediction.
In Handbook of Statistical genetics (eds. Balding D. et al.), John Wiley & Sons, Ltd., p. 83-127.
3. Solovyev VV, Shahmuradov IA. (2003)
PromH: Promoters identification using orthologous genomic sequences.
Nucleic Acids Res. 31(13):3540-3545.

TSSP output:

First line - name of your sequence;
Second and Third lines - LDF threshold and the length of presented sequence
4th line - The number of predicted promoter regions
Next lines - positions of predicted sites, their 'weights' and TATA box position (if found)
Position shows the first nucleotide of the transcript (TSS position)
After that functional motifs are given for each predicted region; (+) or (-) reflects the direct or complementary chain; Fields like "RSP00004 tagaCACGTaga" mean a particular motif identificator with found similar sequence from the Softberry Regsite-Plant data base.

For example:

```
tssp  Wed Jul 10 02:52:32 EDT 2002
>gi|1902902|dbj|AB001920.1| Oryza sativa (japonica cultivar-group) gene for phos
 Length of sequence-      5871
 Thresholds for TATA+ promoters -  0.02, for TATA-/enhancers -  0.04
     2 promoter/enhancer(s) are predicted
 Promoter Pos:   1522 LDF-  0.13 TATA box at   1488    18.93
 Enhancer Pos:   1597 LDF-  0.12
 Transcription factor binding sites/RegSite DB:
for promoter at position -    1522
  1468 (-) RSP00004     tagaCACGTaga
  1459 (+) RSP00010     cACGTG
  1456 (+) RSP00011     ctccACGTGgt
  1461 (+) RSP00016     caTGCAC
  1468 (-) RSP00016     caTGCAC
  1256 (-) RSP00026     gcttttgaTGACtTcaaacac
  1460 (+) RSP00065     ACGTGgcgc
  1460 (+) RSP00066     ACGTGccgc
  1459 (+) RSP00069     tACGTG
  1341 (+) RSP00071     GACGTC
  1346 (-) RSP00071     GACGTC
  1452 (-) RSP00096     GGTTT
  1432 (+) RSP00129     CACGAC
  1281 (+) RSP00148     CGACG
  1284 (+) RSP00148     CGACG
  1315 (+) RSP00148     CGACG
  1335 (+) RSP00148     CGACG
  1340 (+) RSP00148     CGACG
  1365 (+) RSP00148     CGACG
  1434 (+) RSP00148     CGACG
  1458 (+) RSP00148     CGACG
  1347 (-) RSP00148     CGACG
  1474 (+) RSP00162     ACACccGagctaaccacaac
  1348 (+) RSP00241     CGGTCA
  1387 (+) RSP00339     RTTTTTR
  1264 (-) RSP00397     AGTGGCGG
  1268 (+) RSP00422     ACCGAC
  1459 (+) RSP00423     GACGTG
  1464 (-) RSP00424     CACGTC
  1369 (-) RSP00431     rdygRCRGTTRs
  1278 (-) RSP00432     cVacGGTaGGTgg
  1249 (-) RSP00436     TTGACT
  1260 (+) RSP00463     atttcatggCCGACctgcttttt
  1260 (+) RSP00464     acttgatggCCGACctctttttt
  1260 (+) RSP00465     aatatactaCCGACcatgagttct
  1265 (+) RSP00466     actaCCGACatgagttccaaaaagc
  1440 (+) RSP00469     GNGGTG
  1260 (-) RSP00469     GNGGTG
  1440 (+) RSP00470     GTGGNG
  1263 (-) RSP00470     GTGGNG
  1257 (-) RSP00470     GTGGNG
  1390 (+) RSP00477     TTTAA
  1385 (+) RSP00508     gcaTTTTTatca
  1502 (-) RSP00508     gcaTTTTTatca
  1469 (+) RSP00518     tccctACACgcGtcacaattc
  1465 (+) RSP00519     caattcaggACACgtGccctcttca
  1474 (+) RSP00521     ACACccG
  1474 (+) RSP00523     ACACgcG
  1474 (+) RSP00524     ACACgtG
for promoter at position -    1597
  1468 (-) RSP00004     tagaCACGTaga
  1459 (+) RSP00010     cACGTG
  1456 (+) RSP00011     ctccACGTGgt
  1461 (+) RSP00016     caTGCAC
  1468 (-) RSP00016     caTGCAC
  1460 (+) RSP00065     ACGTGgcgc
  1460 (+) RSP00066     ACGTGccgc
  1459 (+) RSP00069     tACGTG
  1341 (+) RSP00071     GACGTC
  1346 (-) RSP00071     GACGTC
  1452 (-) RSP00096     GGTTT
  1432 (+) RSP00129     CACGAC
  1315 (+) RSP00148     CGACG
  1335 (+) RSP00148     CGACG
  1340 (+) RSP00148     CGACG
  1365 (+) RSP00148     CGACG
  1434 (+) RSP00148     CGACG
  1458 (+) RSP00148     CGACG
  1347 (-) RSP00148     CGACG
  1474 (+) RSP00162     ACACccGagctaaccacaac
```

. . . .

Lower cased letters mean non-conserved nucleotides in the site consensus

The letters except (A,T,G,C) describe ambiguous sites in a given DNA sequence motif, where a single character may represent more than one nucleotide using Standard IUPAC Nucleotide code.

See TABLE at http://www.yeastract.com/help/help_searchbydnamotif.php#Ref1

IUPAC Code	**Meaning**	**Origin of Description**
G	G	Guanine
A	A	Adenine
T	T	Thymine
C	C	Cytosine
R	G or A	puRine
Y	T or C	pYrimidine
M	A or C	aMino
K	G or T	Ketone
S	G or C	Strong interaction
W	A or T	Weak interaction
H	A or C or T	not-G, H follows G in the alphabet
B	G or T or C	not-A, B follows A in the alphabet
V	G or C or A	not-T (not-U), V follows U in the alphabet
D	G or A or T	not-C, D follows C in the alphabet
N	G or A or T or C	aNy

Parameters:

Input	
Sequence	Name of the input file
Output	
Result	Name of the output file

PromH-PL

Search for plant promoters using 2 homologous 5'-regions

Protein Location/Motifs

CTL-Epitope

This program is designed for prediction of CTL epitopes of length=9 in protein sequences.

Datasets

For training data we used set of epitopes of length 9 from MHCBN database (Bhasin *et al*, (2003) *Bioinformatics*, 19,666). CTL epitopes which possess binding and activity and sequence length 9 were selected from the database without non-standard amino acid codes and no sequence duplication.

To construct negative dataset we found all sequences from SWISS-PROT database that contain at least one of the epitopes (1717 sequences). From these sequences all the overlapping fragments of length 9 were obtained. From this set of overlapping peptides those were removed, which overlapped with epitope sequences. The remained sequences were filtered so that any of the pair of sequences have no more tan one amino acid in common out of 9 positions. The epitope sequences (932) are the positive set, all the other sequence fragments comprise the negative set (131710). To test the performance the overall data set was splitted randomly on the training and testing sets. The training set comprises 112380 sequences (704 positive). The testing set comprise of 20262 sequences (228 out of them were positive).

Algorithm

To classify sequences the following scores were implemented. (1) Weight matrix scores for each peptide position for PSSM (position specific scoring matrix) formed by positive set sequences, they presented ; (2) positive and negative sequence sets are scanned for the sequence similarity by BLOSUM62 matrix with query sequence and top 5 sequences from both sets separately is determined (5 top from positive set, 5 top from negative set). The similarity scores for positive set ranked by their value and formed additional 5 classification parameters. The similarity scores for negative set ranked by their value and formed another 5 classification parameters. Overall 19 parameters are implemented (9 PSSM positional weights, 5 top positive set similarity scores and 5 top negative set similarity scores). The separation is performed by Linear Discriminant Analysis.

Error estimates

Error estimates on the test set were calculated:

The prediction quality (fraction of correctly predicted sequences) q=0.839058.
npos=228 (epitope sequences)
npos_true=178
npos_false=50
nneg=20034 (non-epitope sequences)
nneg_true=16823
nneg_false=3211
Quality: all=0.839
Positive set =0.781
Negative set=0.840
Input data:
Protein sequence in 20-letter alphabet in FASTA format.

Input Parameters:

- List Output: if this check box is set checked, output data contain list of predicted peptides with their locations in the sequence and scores.
- Threshold: This parameter specifies at which score value will separate positive examples (predicted epitopes, score >= threshold) and negative examples (non-epitopes, score < threshold). By default, threshold=0 (recommended).

Output data:

For each position of the sequence (except eight C-terminal positions) the program output whether the polypeptide of length 9 starting at this position is predicted as cytotoxic T lymphocyte epitope(*) or not ( ). If List Output checkbox is checked, list of predicted epitopes is printed out.

Output example

```
# CTL-epitope-Finder ver. 1.1:
# Program for prediction of  putative cytotoxic T-lymphocyte (CTL) epitopes
# Softberry Inc., 2005
# N-terminal positions of positive peptides (length=9) marked by '*'
# THRESHOLD=0.000
# SEQUENSE LENGTH=191
# NUMBER OF POSITIVE PREDICTIONS=20
# Epitope prediction:
>HCV_core
    .    10   .    20   .    30   .    40   .    50   .    60
MSTNPKPQKKNNRNTNRRPQDVKFPGGGQIVGGVYLLPRRGPRLGVRATRKTSERSQPRG
*     *                    *       *    * *             *
    .    70   .    80   .    90   .    100  .    110  .    120
RRQPIPKARQPEGRAWAQPGYPWPLYGNEGLGWAGWLLSPRGSRPSWGPTDPRRRSRNLG
 *     *              *    *  *            *
    .    130  .    140  .    150  .    160  .    170  .    180
KVIDTLTCGFADLMGYIPLVGAPLGGAARALAHGVRVLEDGVNYATGNLPGCSFSIFLLA
           *                               *  * *       ***
    .    190  .    200  .    210  .    220  .    230  .    240
LLSCLTIPASA


# Output positive peptide list
# Start-End [score]: SEQUENCE
  1-  9 [+13.193]: MSTNPKPQK
  7- 15 [ +0.630]: PQKKNNRNT
 28- 36 [+24.625]: GQIVGGVYL
 36- 44 [+27.123]: LLPRRGPRL
 41- 49 [+25.420]: GPRLGVRAT
 43- 51 [+24.164]: RLGVRATRK
 57- 65 [ +2.835]: QPRGRRQPI
 62- 70 [ +4.587]: RQPIPKARQ
 68- 76 [ +1.264]: ARQPEGRAW
 83- 91 [ +2.128]: WPLYGNEGL
 88- 96 [+20.329]: NEGLGWAGW
 91- 99 [ +3.308]: LGWAGWLLS
104-112 [ +6.383]: RPSWGPTDP
132-140 [+14.183]: DLMGYIPLV
164-172 [ +1.569]: YATGNLPGC
167-175 [ +1.402]: GNLPGCSFS
169-177 [+25.489]: LPGCSFSIF
177-185 [ +5.293]: FLLALLSCL
178-186 [ +5.299]: LLALLSCLT
```

```
179-187 [ +1.837]: LALLSCLTI
```

Parameters:

Input	
Sequence	Input file with protein sequence in 20-letter alphabet in FASTA format.
Output	
Result	Output file.
Format	Output format: **Provide list of predicted epitopes** **Don`t provide list of epitopes**
Output	
Threshold	Threshold for epitope/non-epitope classification.

Protcomp-AN

Program for Identification of sub-cellular localization of Eukaryotic proteins: Animal/Fungi.

Protcomp-AN combines several methods of protein localization prediction - neural networks-based prediction; direct comparison with updated base of homologous proteins of known localization; comparisons of pentamer distributions calculated for query and DB sequences; prediction of certain functional peptide sequences, such as signal peptides, signal-anchors, GPI-anchors, transit peptides of mitochondria and chloroplasts and transmembrane segments; and search for certain localization-specific motifs. It means that the program treats correctly complete sequences only, containing signal sequences, anchors, and other functional peptides, if any. The program includes separately trained recognizers for plant proteins, which dramatically improves recognition accuracy. The following table provides approximate prediction accuracy for each compartment of animal/fungal proteins. Testing was performed on a samples of proteins of known localization (~200 in each localization), which were NOT included in training samples for the programs.

Compartment	Percent predicted correctly		
	ver. 4	ver. 5	ver. 6
Nucleus	80	88	91
Plasma Membrane	80	87	100
Extracellular	69	83	86
Cytoplasm	46	63	88
Mitochondria	76	82	89
Endoplasmic Reticulum	67	83	89
Peroxisome	95	97	91
Lysosome	69	91	100
Golgi	57	77	91

Output sample for complete version:

```
ProtComp Version 6. Identifying sub-cellular location (Animals&Fungi)
Seq name: QUERY, Length=376
Significant similarity in Location DB -  Location:Cytoplasmic
Database sequence: AC=P08319 Location:Cytoplasmic  DE  Alcohol dehydrogenase
class II pi chain precurs
Score=14845, Sequence length=391, Alignment length=365
Predicted by Neural Nets - Extracellular (Secreted) with score    2.4
Integral Prediction of protein location: Cytoplasmic with score   14.7
Location weights:     LocDB / PotLocDB / Neural Nets / Pentamers / Integral
Nuclear                 0.0 /      0.0 /        0.71 /      0.00 /     0.71
```

```
Plasma membrane         0.0 /       0.0 /          0.73 /       0.00 /      0.73
Extracellular           0.0 /       0.0 /          2.42 /       0.00 /      2.42
Cytoplasmic         14845.0 /   18465.0 /          0.83 /       8.50 /     14.68
Mitochondrial           0.0 /       0.0 /          0.70 /       0.00 /      0.70
Endoplasm. retic.       0.0 /       0.0 /          0.70 /       0.50 /      1.21
Peroxisomal             0.0 /       0.0 /          0.49 /       0.00 /      0.49
Lysosomal               0.0 /       0.0 /          0.33 /       0.00 /      0.33
Golgi                   0.0 /       0.0 /          0.40 /       0.00 /      0.40
```

LocDB are scores based on query protein's homologies with proteins of known localization.

PotLocDB are scores based on homologies with proteins which locations are not experimentally known but are assumed based on strong theoretical evidence.

Neural Nets are scores have been assigned by neural networks.

Pentamers are scores based on comparisons of pentamer distributions calculated for QUERY and DB sequences.

Integral are final scores as combinations of previous four scores.

In this reduced version time and disk space consuming processes of DB search and comparisons of pentamers' distributions are abandoned. Columns "LocDB" and "PotLocDB" (results of DB search) and/or "Pentamers" (results of comparisons of pentamers' distributions) are excluded from output tables. However, one should remember, that such abandonment decreases recognition accuracy.

While interpreting output results, it must be kept in mind that:

1. Protcomp's scores *per se*, being weights of complex neural networks, do not represent probabilities of protein's location in a particular compartment.
2. Significant homology with protein of known location is a very strong indicator of query protein's location.
3. For neural networks scores, their relative values for different compartments are more important than absolute values, i.e. if the second best score is much lower than the best one, prediction is more reliable, regardless of absolute values.
4. If both neural networks and homology predictions point to the same compartment, this is very reliable prediction.

In this version comparison with base of homologous proteins of known localization as well as comparisons of pentamer distributions calculated for query and DB sequences are absent.

Parameters:

Input	
Sequence	Input file with protein sequence in FASTA format.
Output	
Result	Output file.

ProtcompDB-AN

Program for Identification of sub-cellular localization of Eukaryotic proteins: Animal/Fungi.

ProtcompDB-AN combines several methods of protein localization prediction - neural networks-based prediction; direct comparison with updated base of homologous proteins of known localization; comparisons of pentamer distributions calculated for query and DB sequences; prediction of certain functional peptide sequences, such as signal peptides, signal-anchors, GPI-anchors, transit peptides of mitochondria and chloroplasts and transmembrane segments; and search for certain localization-specific motifs. It means that the program treats correctly complete sequences only, containing signal sequences, anchors, and other functional peptides, if any. The program includes separately trained recognizers for plant proteins, which dramatically improves recognition accuracy. The following table provides approximate prediction accuracy for each compartment of animal/fungal proteins. Testing was performed on a samples of proteins of known localization (~200 in each localization), which were NOT included in training samples for the programs.

Compartment	Percent predicted

	correctly		
	ver. 4	ver. 5	ver. 6
Nucleus	80	88	91
Plasma Membrane	80	87	100
Extracellular	69	83	86
Cytoplasm	46	63	88
Mitochondria	76	82	89
Endoplasmic Reticulum	67	83	89
Peroxisome	95	97	91
Lysosome	69	91	100
Golgi	57	77	91

Output sample for complete version:

```
ProtComp Version 6. Identifying sub-cellular location (Animals&Fungi)
Seq name: QUERY, Length=376
Significant similarity in Location DB -  Location:Cytoplasmic
Database sequence: AC=P08319 Location:Cytoplasmic  DE  Alcohol dehydrogenase
class II pi chain precurs
Score=14845, Sequence length=391, Alignment length=365
Predicted by Neural Nets - Extracellular (Secreted) with score    2.4
Integral Prediction of protein location: Cytoplasmic with score   14.7
Location weights:     LocDB / PotLocDB / Neural Nets / Pentamers / Integral
Nuclear                 0.0 /      0.0 /        0.71 /      0.00 /     0.71
Plasma membrane         0.0 /      0.0 /        0.73 /      0.00 /     0.73
Extracellular           0.0 /      0.0 /        2.42 /      0.00 /     2.42
Cytoplasmic         14845.0 /  18465.0 /        0.83 /      8.50 /    14.68
Mitochondrial           0.0 /      0.0 /        0.70 /      0.00 /     0.70
Endoplasm. retic.       0.0 /      0.0 /        0.70 /      0.50 /     1.21
Peroxisomal             0.0 /      0.0 /        0.49 /      0.00 /     0.49
Lysosomal               0.0 /      0.0 /        0.33 /      0.00 /     0.33
Golgi                   0.0 /      0.0 /        0.40 /      0.00 /     0.40
```

LocDB are scores based on query protein's homologies with proteins of known localization. PotLocDB are scores based on homologies with proteins which locations are not experimentally known but are assumed based on strong theoretical evidence.

Neural Nets are scores have been assigned by neural networks.

Pentamers are scores based on comparisons of pentamer distributions calculated for QUERY and DB sequences.

Integral are final scores as combinations of previous four scores.

To speed up the recognition, a user may optionally abandon time consuming processes of DB search and comparisons of pentamers' distributions using appropriate marks. In these cases columns "LocDB" and "PotLocDB" (results of DB search) and/or "Pentamers" (results of comparisons of pentamers' distributions) are excluded from output tables. However, one should remember, that such abandonment will decrease recognition accuracy.

While interpreting output results, it must be kept in mind that:

1. Protcomp's scores *per se*, being weights of complex neural networks, do not represent probabilities of protein's location in a particular compartment.
2. Significant homology with protein of known location is a very strong indicator of query protein's location.
3. For neural networks scores, their relative values for different compartments are more important than absolute values, i.e. if the second best score is much lower than the best one, prediction is more reliable, regardless of absolute values.
4. If both neural networks and homology predictions point to the same compartment, this is very reliable prediction.

In this version comparison with base of homologous proteins of known localization as well as comparisons o f pentamer distributions calculated for query and DB sequences are absent.

Protcomp-B

Program for Identification of sub-cellular localization of bacterial proteins.

Protcomp-B combines several methods of protein localization prediction - Linear Discriminant Function-based prediction; direct comparison with bases of homologous proteins of known localization; comparisons of pentamer distributions calculated for query and DB sequences; prediction of certain functional peptide sequences, such as signal peptides and transmembrane segments. It means that the program treats correctly complete sequences only, containing signal sequences, anchors, and other functional peptides, if any.

For Gramm-positive bacteria proteins three locations are discriminated: Cytoplasmic, Membrane and Extracellular (Secreted).

For Gramm-negative bacteria proteins five locations are discriminated: Cytoplasmic, Membrane (Outer and Inner), Periplasmic and Extracellular (Secreted).

If bacteria type is not defined locations for Gramm-negative bacteria are discriminated.

Output sample for complete version:

```
ProtComp  Version  3.  Identifying  sub-cellular  location  Bacterial  (Gramm
negative)

Seq name: Test sequence 330
Significant similarity in Location DB -  Location:Membrane
Database sequence: AC=P55569 Location:Membrane   DE   PROBABLE ABC TRANSPORTER
PERMEASE PROTEIN Y4MJ.
Score=16110, Sequence length=333, Alignment length=330
Predicted by LDA staff - Inner Membrane with score    1.4
******** Signal 1-25 is found
******** Transmembrane segments are found: .+59:157-..-174:199+..+225:327+.
Integral Prediction of protein location: Inner Membrane with score    7.0
Location weights:     LocDB / PotLocDB /     LDA     / Pentamers / Integral
 Cytoplasmic           0.00 /     0.00 /        0.02 /      0.00 /     0.02
 Membrane          16110.00 /  4010.00 /        1.42 /      1.51 /     6.95
 Periplasmic           0.00 /     0.00 /       -0.65 /      0.00 /    -0.65
 Secreted              0.00 /     0.00 /        0.08 /      0.03 /     0.10
```

LocDB are scores based on query protein's homologies with proteins of known localization.

PotLocDB are scores based on homologies with proteins which locations are not experimentally known but are assumed based on strong theoretical evidence.

LDA are scores have been assigned by Linear discriminant functions.

Pentamers are scores based on comparisons of pentamer distributions calculated for QUERY and DB sequences.

Integral are final scores as combinations of previous scores.

In this reduced version time and disk space consuming processes of DB search and comparisons of pentamers' distributions are abandoned. Columns "LocDB" and "PotLocDB" (results of DB search) and/or "Pentamers" (results of comparisons of pentamers' distributions) are excluded from output tables. However, one should remember, that such abandonment decreases recognition accuracy.

While interpreting output results, it must be kept in mind that:

1. Protcomp's scores *per se*, being weights of complex functions, do not represent probabilities of protein's location in a particular compartment.

2. Significant homology with protein of known location is a very strong indicator of query protein's location.

3. For LDA scores, their relative values for different compartments are more important than absolute values, i.e. if the second best score is much lower than the best one, prediction is more reliable, regardless of absolute values.

4. If both LDA and other predictions point to the same compartment, this is very reliable prediction.

In this version comparison with base of homologous proteins of known localization as well as comparisons of pentamer distributions calculated for query and DB sequences are absent.

Parameters:

Input	
Sequence	Input file with protein sequence in FASTA format.
Output	
Result	Output file.
Options	
ramm-negative/Gramm-positive	Is the protein extracted from Gramm-negative or Gramm-positive bacteria?: **Gramm-negative** **Gramm-positive**

ProtcompDB-B

Program for Identification of sub-cellular localization of bacterial proteins.

ProtcompDB-B combines several methods of protein localization prediction - Linear Discriminant Function-based prediction; direct comparison with bases of homologous proteins of known localization; comparisons of pentamer distributions calculated for query and DB sequences; prediction of certain functional peptide sequences, such as signal peptides and transmembrane segments. It means that the program treats correctly complete sequences only, containing signal sequences, anchors, and other functional peptides, if any.

For Gramm-positive bacteria proteins three locations are discriminated: Cytoplasmic, Membrane and Extracellular (Secreted).

For Gramm-negative bacteria proteins five locations are discriminated: Cytoplasmic, Membrane (Outer and Inner), Periplasmic and Extracellular (Secreted).

If bacteria type is not defined locations for Gramm-negative bacteria are discriminated.

Output sample for complete version:

```
ProtComp Version 3. Identifying sub-cellular location Bacterial (Gramm
negative)

Seq name: Test sequence 330
Significant similarity in Location DB -  Location:Membrane
Database sequence: AC=P55569 Location:Membrane  DE  PROBABLE ABC TRANSPORTER
PERMEASE PROTEIN Y4MJ.
Score=16110, Sequence length=333, Alignment length=330
Predicted by LDA staff - Inner Membrane with score    1.4
******** Signal 1-25 is found
******** Transmembrane segments are found: .+59:157-..-174:199+..+225:327+.
Integral Prediction of protein location: Inner Membrane with score    7.0
Location weights:     LocDB / PotLocDB /     LDA     / Pentamers / Integral
 Cytoplasmic           0.00 /     0.00 /        0.02 /      0.00 /     0.02
 Membrane          16110.00 /  4010.00 /        1.42 /      1.51 /     6.95
 Periplasmic           0.00 /     0.00 /       -0.65 /      0.00 /    -0.65
 Secreted              0.00 /     0.00 /        0.08 /      0.03 /     0.10
```

LocDB are scores based on query protein's homologies with proteins of known localization.

PotLocDB are scores based on homologies with proteins which locations are not experimentally known but are assumed based on strong theoretical evidence.

LDA are scores have been assigned by Linear discriminant functions.

Pentamers are scores based on comparisons of pentamer distributions calculated for QUERY and DB sequences.

Integral are final scores as combinations of previous scores.

To speed up the recognition, a user may optionally abandon time consuming processes of DB search and comparisons of pentamers' distributions using appropriate marks. In these cases columns "LocDB" and "PotLocDB" (results of DB search) and/or "Pentamers" (results of comparisons of pentamers' distributions) are excluded from output tables. However, one should remember, that such abandonment will decrease recognition accuracy.

While interpreting output results, it must be kept in mind that:

1. Protcomp's scores *per se*, being weights of complex functions, do not represent probabilities of protein's location in a particular compartment.
2. Significant homology with protein of known location is a very strong indicator of query protein's location.
3. For LDA scores, their relative values for different compartments are more important than absolute values, i.e. if the second best score is much lower than the best one, prediction is more reliable, regardless of absolute values.
4. If both LDA and other predictions point to the same compartment, this is very reliable prediction.

In this version comparison with base of homologous proteins of known localization as well as comparisons of pentamer distributions calculated for query and DB sequences are absent.

Protcomp-PL

Program for Identification of sub-cellular localization of Eukaryotic proteins: Plants

Protcomp combines several methods of protein localization prediction - neural networks-based prediction; direct comparison with updated base of homologous proteins of known localization; comparisons of pentamer distributions calculated for query and DB sequences; prediction of certain functional peptide sequences, such as signal peptides, signal-anchors, GPI-anchors, transit peptides of mitochondria and chloroplasts and transmembrane segments; and search for certain localization-specific motifs. It means that the program treats correctly complete sequences only, containing signal sequences, anchors, and other functional peptides, if any. The program includes separately trained recognizers for animal/fungal and plant proteins, which dramatically improves recognition accuracy. The following table provides approximate prediction accuracy for each compartment of animal/fungal proteins. Testing was performed on a samples of proteins of known localization (~200 in each localization), which were NOT included in training samples for the programs.

Compartment	Percent predicted correctly		
	ver. 4	ver. 5	ver. 6
Nucleus	80	88	91
Plasma Membrane	80	87	100
Extracellular	69	83	86
Cytoplasm	46	63	88
Mitochondria	76	82	89
Endoplasmic Reticulum	67	83	89
Peroxisome	95	97	91
Lysosome	69	91	100
Golgi	57	77	91

Output sample for complete version:

```
Seq name: Q7M1E7 Location:Extracellular (Secreted)  DE  Polygalacturonase
precursor (PG) 514
Significant similarity in Location DB -  Location:Extracellular (Secreted)
Database  sequence:  AC=P35336  Location:Extracellular  (Secreted)     DE
Polygalacturonase precursor (EC 3.
Score=7765, Sequence length=467, Alignment length=335
```

```
Predicted by Neural Nets - Extracellular (Secreted) with score    2.7
******** Signal 1-49 is found
Integral Prediction of protein location: Extracellular (Secreted) with score
4.4
Location weights:     LocDB / PotLocDB / Neural Nets / Pentamers / Integral
 Nuclear                0.0 /      0.0 /        0.70 /      0.08 /     0.77
 Plasma membrane        0.0 /      0.0 /        1.06 /      4.36 /     5.42
 Extracellular       7765.0 /      0.0 /        2.68 /      0.00 /     4.41
 Cytoplasmic            0.0 /      0.0 /        0.72 /      0.00 /     0.72
 Mitochondrial          0.0 /      0.0 /        0.70 /      0.00 /     0.70
 Chloroplast            0.0 /      0.0 /        0.65 /      0.00 /     0.65
 Endoplasm. retic.      0.0 /      0.0 /        1.58 /      0.00 /     1.58
 Peroxisomal            0.0 /      0.0 /        0.48 /      0.00 /     0.48
```

LocDB are scores based on query protein's homologies with proteins of known localization.

PotLocDB are scores based on homologies with proteins which locations are not experimentally known but are assumed based on strong theoretical evidence.

Neural Nets are scores have been assigned by neural networks.

Pentamers are scores based on comparisons of pentamer distributions calculated for QUERY and DB sequences.

Integral are final scores as combinations of previous four scores.

In this reduced version time and disk space consuming processes of DB search and comparisons of pentamers' distributions are abandoned. Columns "LocDB" and "PotLocDB" (results of DB search) and/or "Pentamers" (results of comparisons of pentamers' distributions) are excluded from output tables. However, one should remember, that such abandonment decreases recognition accuracy.

While interpreting output results, it must be kept in mind that:

1. Protcomp's scores *per se*, being weights of complex neural networks, do not represent probabilities of protein's location in a particular compartment.
2. Significant homology with protein of known location is a very strong indicator of query protein's location.
3. For neural networks scores, their relative values for different compartments are more important than absolute values, i.e. if the second best score is much lower than the best one, prediction is more reliable, regardless of absolute values.
4. If both neural networks and homology predictions point to the same compartment, this is very reliable prediction.

In this version comparison with base of homologous proteins of known localization as well as comparisons of pentamer distributions calculated for query and DB sequences are absent.

Parameters:

	Input
Sequence	Input file with protein sequence in FASTA format.
	Output
Result	Output file.

ProtcompDB-PL

Program for Identification of sub-cellular localization of Eukaryotic proteins: Plants.

ProtcompDB-PL combines several methods of protein localization prediction - neural networks-based prediction; direct comparison with updated base of homologous proteins of known localization; comparisons of pentamer distributions calculated for query and DB sequences; prediction of certain functional peptide sequences, such as signal peptides, signal-anchors, GPI-anchors, transit peptides of mitochondria and chloroplasts and transmembrane segments; and search for certain localization-specific motifs. It means that the program treats correctly complete sequences only, containing signal sequences, anchors, and other functional peptides, if any. The program includes separately trained recognizers for animal/fungal and plant proteins, which dramatically improves recognition accuracy. The following table provides approximate prediction accuracy for each compartment of animal/fungal proteins. Testing was performed on a samples of proteins of known localization (~200 in each localization), which were NOT included in training samples for the programs.

Compartment	Percent predicted correctly		
	ver. 4	ver. 5	ver. 6
Nucleus	80	88	91
Plasma Membrane	80	87	100
Extracellular	69	83	86
Cytoplasm	46	63	88
Mitochondria	76	82	89
Endoplasmic Reticulum	67	83	89
Peroxisome	95	97	91
Lysosome	69	91	100
Golgi	57	77	91

Output sample for complete version:

```
Seq name: Q7M1E7 Location:Extracellular (Secreted)   DE   Polygalacturonase
precursor (PG) 514
Significant similarity in Location DB -  Location:Extracellular (Secreted)
Database   sequence:   AC=P35336   Location:Extracellular   (Secreted)      DE
Polygalacturonase precursor (EC 3.
Score=7765, Sequence length=467, Alignment length=335
Predicted by Neural Nets - Extracellular (Secreted) with score    2.7
******** Signal 1-49 is found
Integral Prediction of protein location: Extracellular (Secreted) with score
4.4
Location weights:     LocDB / PotLocDB / Neural Nets / Pentamers / Integral
 Nuclear                0.0 /      0.0 /        0.70 /      0.08 /     0.77
 Plasma membrane        0.0 /      0.0 /        1.06 /      4.36 /     5.42
 Extracellular       7765.0 /      0.0 /        2.68 /      0.00 /     4.41
 Cytoplasmic            0.0 /      0.0 /        0.72 /      0.00 /     0.72
 Mitochondrial          0.0 /      0.0 /        0.70 /      0.00 /     0.70
 Chloroplast            0.0 /      0.0 /        0.65 /      0.00 /     0.65
 Endoplasm. retic.      0.0 /      0.0 /        1.58 /      0.00 /     1.58
 Peroxisomal            0.0 /      0.0 /        0.48 /      0.00 /     0.48
```

LocDB are scores based on query protein's homologies with proteins of known localization.

PotLocDB are scores based on homologies with proteins which locations are not experimentally known but are assumed based on strong theoretical evidence.

Neural Nets are scores have been assigned by neural networks.

Pentamers are scores based on comparisons of pentamer distributions calculated for QUERY and DB sequences.

Integral are final scores as combinations of previous four scores.

To speed up the recognition, a user may optionally abandon time consuming processes of DB search and comparisons of pentamers' distributions using appropriate marks. In these cases columns "LocDB" and "PotLocDB" (results of DB search) and/or "Pentamers" (results of comparisons of pentamers' distributions) are excluded from output tables. However, one should remember, that such abandonment will decrease recognition accuracy.

While interpreting output results, it must be kept in mind that:

1. Protcomp's scores *per se*, being weights of complex neural networks, do not represent probabilities of protein's location in a particular compartment.

2. Significant homology with protein of known location is a very strong indicator of query protein's location.

3. For neural networks scores, their relative values for different compartments are more important than absolute values, i.e. if the second best score is much lower than the best one, prediction is more reliable, regardless of absolute values.
4. If both neural networks and homology predictions point to the same compartment, this is very reliable prediction.
In this version comparison with base of homologous proteins of known localization as well as comparisons of pentamer distributions calculated for query and DB sequences are absent.

PSite

Search for of prosite patterns with statistical estimation
Method description:
The method is based on statistical estimation of expected number of a prosite pattern in a given sequence. It uses the PROSITE database (author: Amos Bairoch,1995) of functional motifs. If we found a pattern which has expected number significantly less than 1, it can be supposed that the analyzed sequence possesses the pattern function. Presented version 1 is the simplest version that search for patterns without any deviation from a given Prosite consensus. In the following version we will include this possibility. In the output of PSite we can see a prosite pattern, its position in the sequence, accession number, ID, Description in the PROSITE database as well as Document number where is pattern characteristics outlined. It must be noted that patterns which started at the beginning or end of protein sequence will be recognized along the whole sequence in this version. It may be useful for analysis of ORF or 6 frame translation sequences.
Input sequence for this program should be in fasta format with 80 or less sequence letters per line.
Acknowledgments: We acknowledge Ilgam Shahmuradov and Igor Rogozin which took part in development some applications of this method for nucleotide consensuses searching and Asya Salihova for protein sites searching on IBM PC.
Example of PSite output:

```
 PSite V1 - search for Prosite patterns
        10         20         30         40         50         60
 RLLRAIMGAPGSGKGTVSSRITKHFELKHLSSGDLLRDNMLRGTEIGVLAKTFIDQGKLI
        70         80         90        100        110        120
 PDDVMTRLVLHELKN*TQYNWLLDGFPRTLPQAEALDRAYQIDTVINLNVPFEVIKQRLT
       130        140        150        160        170        180
 ARWIHPGSGRVYNIEFNPPKTMGIDDLTGEPLVQREDDRPETVVKRLKAYEAQTEPVLEY
       190        200        210        220        230        240
 YRKKGVLETFSYTETNKIWPHVYAFLQTKLPDANKDDALDQREWSAAAAWLAAAAALDLN
       250        260        270        280        290        300
 AGCPAAALAAAAAGSAACAAAAAFAAAAAACCAACAAAAAAAACAAAADAACGAYAYACAP

ID   GLYCOSAMINOGLYCAN; RULE.
AC   PS00002;
DE   Glycosaminoglycan attachment site.
DO   PDOC00002;
PA   S-G-x-G.
 Sites found:  1 Expected number:   0.0272 95% confidential interval:   0
  #  Start  End  Expected  Site sequence
  1    12    15   0.0272   SGKG
ID   EF_HAND; PATTERN.
AC   PS00018;
DE   EF-hand calcium-binding domain.
DO   PDOC00018;
PA   D-x-[DNS]-{ILVFYW}-[DENSTG]-[DNQGHRK]-{GP}-[LIVMC]-[DENQSTAGC]-x(2)-
PA   [DE]-[LIVMFYW].
 Sites found:  1 Expected number:   0.0004 95% confidential interval:   0
  #  Start  End  Expected  Site sequence
  1   212   224   0.0004   DANKDDALDQREW
ID   ADENYLATE_KINASE; PATTERN.
```

```
AC   PS00113;
DE   Adenylate kinase signature.
DO   PDOC00104;
PA   [LIVMFYW](3)-D-G-[FY]-P-R-x(3)-[NQ].
 Sites found:  1 Expected number:   0.0000 95% confidential interval:   0
  #  Start  End  Expected  Site sequence
  1    81    92   0.0000  WLLDGFPRTLPQ
```

Reference:

Solovyev V.V., Kolchanov N.A. 1994,
Search for functional sites using consensus
In Computer analysis of Genetic macromolecules. (eds. Kolchanov N.A., Lim H.A.), World Scientific, p.16-21.

Parameters:

	Input
Sequence	Input file with protein sequence in 20-letter alphabet in FASTA format.
	Output
Result	Output file.

Protein Structure

3D-Comp

3D-Comp is intended for superposing tertiary structures of two proteins basing on alignment of their primary sequences.

Input data:
PDB file with the structure of protein 1;
PDB file with the structure of protein 2; and
Alignment of these protein sequences.
Output data:
PDB file with superposed structures;
RMSD of C-alpha atoms; and
Location parameters and rotation matrix.
Algorithm:
The method of best superposition of spatial structures independent of their initial positions in the space (Kabsch, 1976) was realized.
Location parameters and rotation matrix are calculated according to C-alpha atoms.
Reference:
Kabsch W. A solution for the best rotation to relate two sets of vectors. Acta Cryst. 1976; A32: 922-923.
Output example:

```
HEADER    PROTEIN STRUCTURE ALIGNMENT
COMPND    (A) file1 chain A (B) file2 chain B
REMARK   1
REMARK   1 Transformation of chain A coordinates:
REMARK   1 Anew = U*(Aold-shift1)+shift2
REMARK   1 The rotation matrix U:
REMARK   1       0.2843  0.9037  0.3184
REMARK   1      -0.3886 -0.1940  0.9003
REMARK   1       0.8767 -0.3809  0.2969
REMARK   1
REMARK   1 shift1 (X, Y, Z) = ( 24.434,   9.342,   8.358)
REMARK   1 shift2 (X, Y, Z) = ( 25.967,  64.677,  13.625)
REMARK   1
REMARK   1 RMSD on Ca-atoms:  3.684 angstrom
REMARK   1
ATOM      1  N   MET A   1      38.730  55.215  -3.247  1.00  0.00
ATOM      2  CA  MET A   1      38.092  55.938  -2.140  1.00  0.00
ATOM      3  C   MET A   1      36.924  56.821  -2.592  1.00  0.00
ATOM      4  O   MET A   1      37.119  57.872  -3.206  1.00  0.00
ATOM      5  CB  MET A   1      39.133  56.786  -1.392  1.00  0.00
ATOM      6  CG  MET A   1      38.587  57.621  -0.216  1.00  0.00
ATOM      7  SD  MET A   1      37.784  56.643   1.092  1.00  0.00
ATOM      8  CE  MET A   1      39.147  56.452   2.275  1.00  0.00
ATOM      9  N   GLN A   2      35.708  56.384  -2.279  1.00  0.00
ATOM     10  CA  GLN A   2      34.509  57.134  -2.635  1.00  0.00
ATOM     11  C   GLN A   2      33.808  57.700  -1.397  1.00  0.00
ATOM     12  O   GLN A   2      34.004  57.211  -0.285  1.00  0.00
ATOM     13  CB  GLN A   2      33.546  56.247  -3.414  1.00  0.00
ATOM     14  CG  GLN A   2      34.062  55.820  -4.780  1.00  0.00
ATOM     15  CD  GLN A   2      33.012  55.077  -5.594  1.00  0.00
ATOM     16  OE1 GLN A   2      31.804  55.288  -5.421  1.00  0.00
ATOM     17  NE2 GLN A   2      33.468  54.204  -6.493  1.00  0.00
ATOM     18  N   THR A   3      32.998  58.738  -1.593  1.00  0.00
ATOM     19  CA  THR A   3      32.277  59.357  -0.488  1.00  0.00
ATOM     20  C   THR A   3      30.778  59.069  -0.511  1.00  0.00
ATOM     21  O   THR A   3      30.168  58.918  -1.578  1.00  0.00
ATOM     22  CB  THR A   3      32.488  60.881  -0.457  1.00  0.00
ATOM     23  OG1 THR A   3      33.891  61.165  -0.440  1.00  0.00
ATOM     24  CG2 THR A   3      31.844  61.495   0.797  1.00  0.00
ATOM     25  N   ILE A   4      30.215  58.923   0.686  1.00  0.00
ATOM     26  CA  ILE A   4      28.785  58.693   0.871  1.00  0.00
ATOM     27  C   ILE A   4      28.292  59.883   1.697  1.00  0.00
ATOM     28  O   ILE A   4      28.614  59.996   2.881  1.00  0.00
ATOM     29  CB  ILE A   4      28.490  57.386   1.652  1.00  0.00
      ……………………..
ATOM   2962  CB  LEU B 385       7.514  70.764 -17.815  1.00  0.00
ATOM   2963  CG  LEU B 385       7.267  70.676 -16.308  1.00  0.00
ATOM   2964  CD1 LEU B 385       6.707  71.973 -15.753  1.00  0.00
ATOM   2965  CD2 LEU B 385       6.317  69.529 -15.982  1.00  0.00
ATOM   2966  N   SER B 386       9.587  69.697 -20.509  1.00  0.00
ATOM   2967  CA  SER B 386       9.716  69.739 -21.951  1.00  0.00
ATOM   2968  C   SER B 386      10.554  70.875 -22.532  1.00  0.00
ATOM   2969  O   SER B 386      10.781  71.899 -21.850  1.00  0.00
ATOM   2970  OXT SER B 386      10.967  70.744 -23.728  1.00  0.00
```

Parameters:

Input	
PDB structure 1	First structure file name
PDB structure 2	Second structure file name
Input format 1	First structure file format
Input format 2	Second structure file format
Structure 1 chain ID	First structure chain ID
Structure 2 chain ID	Second structure chain ID
Alignment	File with sequences aligment in FASTA format.
Output	
Result	Name of the output file.

3D-Match

3D-Match implements pairwise protein structure alignment.

The algorithm implements a three-step procedure for aligning protein three-dimensional structures. The procedure includes building of the alignment core with the optimal RMSD, its expansion by introducing new protein fragments into the alignment, and optimization using dynamic programming to finally achieve an optimal alignment. 3D-Match aligns two polypeptide chains using C-alpha atomic coordinates, secondary structure characteristics are additionally used to weight the alignment.

The input is the PDB file and the polypeptide chain identifier for each protein of a queried pair. In the case when the chain identifier is not provided, a protein structure comparison is performed using the first polypeptide chain found in the protein.

Output data.

Structural alignment is represented in PDB format in which the queried structures are assigned different chain IDs. The values for the RMSD, Zscore and structure-based sequence alignment are accommodated in the REMARK field.

Zscore is a measure of the statistical significance of the structural alignment of the queried proteins relative to an alignment of random structures. As a rule, the score for proteins with a similar fold will be 3.5, even better than that.

An example of output data.

```
HEADER    PROTEIN STRUCTURE ALIGNMENT
COMPND    (A) 1BWW chain A (B) 2BFV chain L
```

```
REMARK   1
REMARK   1 RMSD on Ca-atoms:  0.791 angstrom
REMARK   1 Zscore          :  6.230
REMARK   1
REMARK   1
REMARK   1 Alignment
REMARK   1
REMARK   1 3     DIQMTQSPSSLSASVGDRVTITCQASQDII-----KYLNWYQQKPGKAPKLLIYEASNLQ
REMARK   1 1     DIELTQSPPSLPVSLGDQVSISCRSSQSLVSNNRRNYLHWYLQKPGQSPKLVIYKVSNRF
REMARK   1
REMARK   1 58    AGVPSRFSGSGSGTDYTFTISSLQPEDIATYYCQQYQSLPYTFGQGTKL
REMARK   1 61    SGVPDRFSGSGSGTDFTLKISRVAAEDLGLYFCSQSSHVPLTFGSGTKL
REMARK   1
ATOM      1  N   THR A   1     -18.648   5.701 -17.803  1.00 67.85           N
ATOM      2  CA  THR A   1     -18.151   6.056 -16.472  1.00 64.75           C
ATOM      3  C   THR A   1     -16.630   6.135 -16.463  1.00 48.48           C
ATOM      4  O   THR A   1     -15.942   5.184 -16.867  1.00 47.02           O
ATOM      5  CB  THR A   1     -18.621   5.088 -15.373  1.00 72.33           C
ATOM      6  OG1 THR A   1     -19.566   4.118 -15.842  1.00 76.14           O
ATOM      7  CG2 THR A   1     -19.338   5.863 -14.272  1.00 80.20           C
ATOM      8  N   PRO A   2     -16.032   7.229 -16.013  1.00 34.29           N
ATOM      9  CA  PRO A   2     -14.555   7.266 -16.013  1.00 29.06           C
ATOM     10  C   PRO A   2     -14.037   6.265 -14.977  1.00 29.14           C
ATOM     11  O   PRO A   2     -14.654   6.023 -13.941  1.00 27.39           O
ATOM     12  CB  PRO A   2     -14.217   8.680 -15.566  1.00 28.31           C
ATOM     13  CG  PRO A   2     -15.493   9.424 -15.458  1.00 30.57           C
ATOM     14  CD  PRO A   2     -16.595   8.410 -15.368  1.00 32.32           C
ATOM     15  N   ASP A   3     -12.875   5.683 -15.224  1.00 27.28           N
ATOM     16  CA  ASP A   3     -12.313   4.811 -14.192  1.00 21.41           C
```

Parameters:

Input	
PDB structure 1	First structure file name
PDB structure 2	Second structure file name
Input format 1	First structure file format
Input format 2	Second structure file format
Structure 1 chain ID	First structure chain ID
Structure 2 chain ID	Second structure chain ID
Output	
Result	Output file

3D-MatchDB

3D-MatchDB is a program for searching a database of protein 3D structures for structural homology with a query protein. To improve speed, 3D-MatchDB uses an algorithm of fast alignment of secondary structure elements (helix, beta-sheet) and preprocessed PDB database, which has secondary structure elements mapped to 3D structures. Current version has 12,834 protein chains from PDB, cleared from redundant entries, so that their sequence homologies are not higher than 98%. 3D-MatchDB performs pairwise structural alignment of query protein with each database entry, calculates RMSD, Zscore, Aligned Size, and number of gaps for each alignment, and outputs a sorted list of entries that have structural homology to query protein with RMSD less than 5 angstrom and Zscore above 3.2. Then user can get atomic coordinates of structurally aligned pairs of proteins by picking one structure from that list and using 3D-Match program for refined alignment.

Parameters calculated by 3D-MatchDB (RMSD, Zscore, Aligned Size, and number of gaps) may slightly differ from those calculated by 3D-Match, as the former uses faster and slightly less accurate alignment algorithm.

Input data.

PDB file and identifier of peptide chain for query protein are used as input data. If chain identifier is not provided, alignment is performed for first polypeptide chain found in a protein.

Output data.

User can choose output of structure database search to be sorted by Zscore or by RMSD by checking a corresponding box.

The output is a list of structural homologs, containing PDB identifier, chain identifier, and description from COMPND field of PDB for each protein, as well as RMSD, Zscore, Aligned Size, and number of gaps for alignment of that protein with query one.

To get protein structure alignment, user should check the corresponding line in an output list, and then check "Get structure alignment as text". 3D-Match program will then produce a structural alignment of query and chosen proteins and output it either in text. In case of text output, structural alignment is presented in PDB format with values for RMSD, Zscore and structure-based sequence alignment placed in REMARK field.

Fast comparison of 3D structures.

Fast comparison of 3D structures is based on an algorithm of secondary structure elements alignment, similar to that of 3D-Match, but with slight modifications to improve speed. Detailed description of this algorithm is given in description of 3D-Match program. Modifications concern mostly checking alignment quality on each step of an algorithm. First check is performed upon building a core of alignment. If RMSD is above certain threshold, or contains number of secondary structure elements below threshold, the structure is discarded. Second check is performed during transformation from secondary structure-based alignment to that based on coordinates of Ca atoms.

Presence or absence of structural homology usually becomes evident on the stage of building core alignment. If there is no homology, core would have high RMSD or be very short. Therefore, most PDB entries are discarded at this stage, which dramatically increases speed of PDB search.

```
Example of data output.
STRUCTURE DATABASE SEARCHING.

1BAN:A ZScore= 6.6 RMSD=  0.31 Aligned=108 Size=108 Gaps=0 Name=BARNASE (G SPECIFIC
ENDONUCLEASE) (E.C.3.1.27.-) MUTANT  WITH SER 91 REPLACED BY ALA (S91A)
2RBI:A ZScore= 6.6 RMSD=  0.37 Aligned=108 Size=108 Gaps=0 Name=MOL_ID: 1;  MOLECULE:
RIBONUCLEASE;  CHAIN: A, B;  SYNONYM: BINASE, EXTRACELLULAR RIBONUCLEASE FROM BACILLUS
INTERMEDIUS;  EC: 3.1.27.-;  ENGINEERED: YES;  MUTATION: H101N
1A2P:A ZScore= 6.6 RMSD=  0.00 Aligned=108 Size=108 Gaps=0 Name=MOL_ID: 1;  MOLECULE:
BARNASE;  CHAIN: A, B, C;  EC: 3.1.27.-;  ENGINEERED: YES
1BSB:A ZScore= 6.6 RMSD=  0.17 Aligned=108 Size=108 Gaps=0 Name=BARNASE (G SPECIFIC
ENDONUCLEASE) (E.C.3.1.27.-) MUTANT  WITH ILE 76 REPLACED BY VAL (I76V)
1BNS:A ZScore= 6.6 RMSD=  0.27 Aligned=108 Size=108 Gaps=0 Name=BARNASE (G SPECIFIC
ENDONUCLEASE) (E.C.3.1.27.-) MUTANT  WITH THR 26 REPLACED BY ALA (T26A)
1BNG:A ZScore= 6.6 RMSD=  0.22 Aligned=108 Size=108 Gaps=0 Name=BARNASE (E.C.3.1.27.-)
DISULFIDE MUTANT WITH SER 85  REPLACED BY CYS AND HIS 102 REPLACED BY CYS (S85C,H102C)
1BAO:A ZScore= 6.6 RMSD=  0.20 Aligned=108 Size=108 Gaps=0 Name=BARNASE (G SPECIFIC
ENDONUCLEASE) (E.C.3.1.27.-) MUTANT  WITH TYR 78 REPLACED BY PHE (Y78F)
1BRI:A ZScore= 6.6 RMSD=  0.23 Aligned=107 Size=107 Gaps=1 Name=BARNASE (E.C.3.1.27.-)
MUTANT WITH ILE 76 REPLACED BY ALA  (I76A)
1BRG:A ZScore= 6.6 RMSD=  0.26 Aligned=108 Size=108 Gaps=0 Name=BARNASE (G SPECIFIC
ENDONUCLEASE) (E.C.3.1.27.-) MUTANT  WITH PHE 7 REPLACED BY LEU (F7L)
1B20:A ZScore= 6.6 RMSD=  0.30 Aligned=108 Size=109 Gaps=1 Name=MOL_ID: 1;  MOLECULE:
BARNASE;  CHAIN: A, B, C;  EC: 3.1.27.3;  ENGINEERED: YES;  MUTATION: YES
1BRK:A ZScore= 6.6 RMSD=  0.29 Aligned=108 Size=108 Gaps=0 Name=BARNASE (E.C.3.1.27.-)
MUTANT WITH ILE 96 REPLACED BY ALA  (I96A)
1BSC:A ZScore= 6.6 RMSD=  0.18 Aligned=108 Size=108 Gaps=0 Name=BARNASE (G SPECIFIC
ENDONUCLEASE) (E.C.3.1.27.-) MUTANT  WITH ILE 88 REPLACED BY VAL (I88V)
1BNE:A ZScore= 6.6 RMSD=  0.32 Aligned=107 Size=107 Gaps=1 Name=BARNASE (E.C.3.1.27.-)
DISULFIDE MUTANT WITH ALA 43  REPLACED BY CYS AND SER 80 REPLACED BY CYS (A43C,S80C)

PROTEIN STRUCTURE ALIGNMENT.

HEADER    PROTEIN STRUCTURE ALIGNMENT
COMPND    (A) 1A2P chain A (B) 1BAN chain A
REMARK   1
REMARK   1 RMSD on Ca-atoms :  0.313 angstrom
REMARK   1 Zscore           :  6.580
REMARK   1 Aligned positions:    108
REMARK   1 Gap positions    :      0
REMARK   1 Sequence identity:   99.1 (%)
REMARK   1
REMARK   1 Structure based sequence alignment
REMARK   1
REMARK   1 3     VINTFDGVADYLQTYHKLPDNYITKSEAQALGWVASKGNLADVAPGKSIGGDIFSNREGK
REMARK   1 3     VINTFDGVADYLQTYHKLPDNYITKSEAQALGWVASKGNLADVAPGKSIGGDIFSNREGK
REMARK   1
REMARK   1 63    LPGKSGRTWREADINYTSGFRNSDRILYSSDWLIYKTTDHYQTFTKIR
REMARK   1 63    LPGKSGRTWREADINYTSGFRNSDRILYASDWLIYKTTDHYQTFTKIR
REMARK   1
ATOM      1  N   VAL A   3     -12.310  -8.243   5.307  1.00 47.79           N
ATOM      2  CA  VAL A   3     -11.179  -7.573   4.634  1.00 41.49           C
ATOM      3  C   VAL A   3     -11.019  -6.157   5.156  1.00 34.47           C
ATOM      4  O   VAL A   3     -11.979  -5.382   5.128  1.00 34.84           O
ATOM      5  CB  VAL A   3     -11.383  -7.546   3.117  1.00 42.12           C
ATOM      6  CG1 VAL A   3     -10.536  -6.536   2.420  1.00 38.29           C
ATOM      7  CG2 VAL A   3     -11.154  -8.948   2.527  1.00 45.14           C
ATOM      8  N   ILE A   4      -9.810  -5.789   5.545  1.00 27.18           N
ATOM      9  CA  ILE A   4      -9.587  -4.366   5.973  1.00 24.08           C
ATOM     10  C   ILE A   4      -8.788  -3.683   4.864  1.00 21.31           C
ATOM     11  O   ILE A   4      -7.656  -4.064   4.576  1.00 21.63           O
ATOM     12  CB  ILE A   4      -8.731  -4.385   7.264  1.00 24.83           C
ATOM     13  CG1 ILE A   4      -9.399  -5.210   8.386  1.00 27.01           C
ATOM     14  CG2 ILE A   4      -8.372  -2.999   7.701  1.00 24.93           C
ATOM     15  CD1 ILE A   4      -8.582  -5.279   9.651  1.00 33.25           C
ATOM     16  N   ASN A   5      -9.456  -2.797   4.122  1.00 20.12           N
ATOM     17  CA  ASN A   5      -8.814  -2.164   2.982  1.00 19.67           C
ATOM     18  C   ASN A   5      -9.183  -0.706   2.810  1.00 17.24           C
ATOM     19  O   ASN A   5      -8.956  -0.171   1.716  1.00 17.10           O
ATOM     20  CB  ASN A   5      -9.048  -2.927   1.678  1.00 20.04           C
ATOM     21  CG  ASN A   5     -10.495  -2.771   1.189  1.00 20.89           C
ATOM     22  OD1 ASN A   5     -11.360  -2.364   1.950  1.00 21.76           O
ATOM     23  ND2 ASN A   5     -10.710  -3.053  -0.084  1.00 22.93           N
ATOM     24  N   THR A   6      -9.605  -0.043   3.868  1.00 15.82           N
ATOM     25  CA  THR A   6      -9.917   1.401   3.801  1.00 16.81           C
ATOM     26  C   THR A   6      -8.791   2.237   4.362  1.00 14.04           C
ATOM     27  O   THR A   6      -7.944   1.762   5.098  1.00 14.38           O
ATOM     28  CB  THR A   6     -11.207   1.679   4.628  1.00 17.16           C
ATOM     29  OG1 THR A   6     -11.008   1.226   5.948  1.00 23.19           O
ATOM     30  CG2 THR A   6     -12.404   0.966   4.043  1.00 22.55           C
ATOM     31  N   PHE A   7      -8.801   3.561   4.057  1.00 14.44           N
ATOM     32  CA  PHE A   7      -7.792   4.422   4.634  1.00 14.94           C
```

3D-ModelFit

3DModelFit - program for the estimation of quality of 3D model structure of protein

Program accepts model and real (target) 3D structures of protein in PDB format (indexing of residues in files should be identical). Program calculates their optimal superposition and estimates following scores for model quality estimation:

Model N - number of model residues
Target N - number of target residues
Model NP - number of model residues that presented in target structure
Target NP - number of target residues that presented in model structure
RMS_Buried - RMS for buried area of residues in model and target structure
RMS_Polar_fract - RMS for polar fraction buried of residues in model and target structure
SS_Match - fraction of secondary structure match for residues in model and target structure
LCS_score - LCS_TS score (Zemla A. (2003), Nucleic Acids Res. 31:3370-3374)
GDT_score - GDT_TS score (Zemla A. (2003), Nucleic Acids Res. 31:3370-3374)
CHI1_match - fraction of residues matching their chi1 angle
CHI2_match - fraction of residues matching their chi2 angle
CHI12_match - fraction of residues matching their chi1 and chi2 angles
RMS_CA - RMS on CA atoms.

If 'Output format' is set to "Extended" value, program outputs PDB file with structural superposition of model (chain M) and target (chain T) structures.

Remark fields in output file represent also residue to residue correspondence of model and target structutes, for example:

```
REMARK  50 Structure quality:
REMARK  50 M:  G    D    S    V    E    N    Q    S
REMARK  50 N:  15   16   17   18   19   20   21   22
REMARK  50 T:  -    -    -    -    -    -    q    S
```

where M: model amino acid, N: residue index, T: target amino acid. Missed residues are indicated as gaps ('-'); residues with missed side chains are indicated as small letters.

Detailed description of LCS and GDT scores is also presented in remark fields.

Parameters:

Input	
Model structure file	Model structure file name
Target structure file	Target structure file name
Model input format	Model structure file format
Target input format	Target structure file format
Model chain ID	Model structure chain ID
Target chain ID	Target structure chain ID
Output	
Result	Output file
Formatt	Specifies detailed program output (Model-Target structure superposition).
Options	
Chi angle match threshold	Chi angle match threshold

AbIni3D

AbIni3D - Ab inition folding

Problem: The program is intended for calculating 3D structure of proteins, provided that 3D structures of individual parts (fragments) of the protein are known, while phi and psi angles between the fragments should be found. This problem may arise when constructing a protein structure from fragments, whose structures were obtained using the search for homology of their primary sequences.

Method: The angles are calculated by genetic algorithm. The target optimization function is comprised by two additive contributions: (a) energy of the short-range interaction between the fragments and (b) the energy of phi/psi angles constructed basing on statistics of the angles between fragments of secondary structures in protein 3D structures from PDB database.

Results: Testing using seven natural proteins (with lengths from 58 to 135 aa; each protein consisted of several fragments) demonstrated that the program restores the native structure with a mean accuracy of 5.3.6.7 A. The prediction accuracy depends on individual protein and program operation mode: for three best proteins, the mean value of RMSD between the restored and native structures over ten runs amounted to 1.9, 2.3, and 2.6 A.

HELP in questions and answers on the AbIni3D program

Q: For what purpose the program is intended?

A: For calculating protein spatial structures basing on the fragments of whole structure that can be obtained by use of search for homology.

Q: How are the fragments selected?

A: Fragments of protein sequence (homologous regions) should be selected so that they would completely span the whole sequence of the target protein and, on the other hand, should not overlap. The program joins the fragments into a single chain and by use of genetic algorithm, optimizes phi and psi angles at the sites where the fragments were joined to find the conformation displaying a minimal energy.

Q: What are the launching parameters, input, and output formats?

A: The program has two mandatory parameters and one optional: these are the input COV file, output PDB file, and optional parameter-the number of computing cycles for genetic algorithm (default value, 500).

Q: How the run-time should be selected?

A: This depends on the number of fragments-more fragments require a longer run-time. For example, 50 cycles are sufficient for optimizing two fragments.

Q: What is the input COV format?

A: This is a specialized format for the program in question containing information on the primary structure of the fragments, alignments for covering of the target sequence, and "pieces" of PDB files corresponding to the covering fragments.

Example:

```
=====================================================================================

*****  SET 1 *****
>1NDDB qb=0 pb=25 le=20 Sc=98.9
aaaa         bbbbb
MSANFTDKNGRQSKGVLLLR
IKERVEEKEGIPPQQQRLIY
aaaaaaaaa      bbbbb
ATOM    794  N   ILE B 126      37.162  -0.022  40.293  1.00 12.67           N
ATOM    795  CA  ILE B 126      35.962  -0.674  39.781  1.00 11.72           C
ATOM    796  C   ILE B 126      35.671  -0.073  38.399  1.00 12.39           C
ATOM    797  O   ILE B 126      35.366  -0.799  37.452  1.00 14.47           O
ATOM    798  CB  ILE B 126      34.746  -0.424  40.696  1.00 13.18           C
ATOM    799  CG1 ILE B 126      35.033  -0.951  42.107  1.00 14.02           C
ATOM    800  CG2 ILE B 126      33.499  -1.074  40.094  1.00 15.53           C
ATOM    801  CD1 ILE B 126      33.908  -0.706  43.107  1.00 14.94           C
ATOM    802  N   LYS B 127      35.806   1.249  38.282  1.00 11.60           N
ATOM    803  CA  LYS B 127      35.581   1.929  37.006  1.00 11.37           C

....    ...  ..  ... . ...      ......   .....  ......  .... .....           .

ATOM    964  CZ  TYR B 145      25.681  -2.498  47.587  1.00 17.99           C
ATOM    965  OH  TYR B 145      25.481  -3.704  48.220  1.00 20.22           O
>2PDZA qb=20 pb=31 le=17 Sc=93.1
b
TLAMPSDTNANGDIFGG
KIFKGLAADQTEALFVG
b     aaaa
ATOM    498  N   LYS A  32      -1.097  -3.476  -1.916  1.00  0.00           N
....    ...  ..  ... . ...      ......   .....  ......  .... .....           .
TER
======================================================================================
```

There may be several variants of coverings (SETs); therefore, each new variant starts from the corresponding keyword, for example, "SET 1"; next, "SET 2"; etc.

Q: How is it possible to create a COV file?

A: The file mandatory starts with the keyword "SET" with any number, for example, 1, 2, etc., followed one after another by the "pieces" of spatial structures in PDB format. The fragments are separated from one another by an empty string.

Example: suppose, you want to "disrupt" the native structure of a protein (and you have this structure in PDB format) to test then how it will be restored using this program. For this purpose, copy your PDB file, for example, YourProtein.pdb, into the file with a name, for example, YourProtein.cov, and introduce the corresponding changes:

- Put the text, for example, " SET 1 ", into the first string (it is important that the first string would contain the word SET in capitals) and

- Add empty strings at the points where you want to destroy the protein structure (i.e. break the conformation of the main chain); several breaks (empty strings) are recommended, for example, tree-five.

```
 Example:
******* SET 1 *******
REMARK   MSI WebLab Viewer PDB file
REMARK   Created:  Fri Oct 25 07:58:42 ‡€њтр’™ьњ’  Lћ›  (ћ›˜’) 2002
CRYST1   57.810   29.700  106.090  90.00 101.99  90.00 A2
ATOM      1  N   GLY A   1      15.740  11.178 -11.733  1.00  0.00
ATOM      2  CA  GLY A   1      15.234  10.462 -10.556  1.00  0.00
ATOM      3  C   GLY A   1      16.284   9.483  -9.998  1.00  0.00
ATOM      4  O   GLY A   1      17.150   8.979 -10.709  1.00  0.00
....    ...  ..  ... . ...      ......   .....  ......  .... .....
ATOM    310  N   LEU A  40       6.658  -4.909  19.830  1.00  0.00
ATOM    311  CA  LEU A  40       6.751  -5.839  20.961  1.00  0.00
ATOM    312  C   LEU A  40       5.510  -6.747  21.050  1.00  0.00
ATOM    313  O   LEU A  40       5.642  -7.969  21.132  1.00  0.00
ATOM    314  CB  LEU A  40       6.968  -5.086  22.286  1.00  0.00
ATOM    315  CG  LEU A  40       7.926  -5.898  23.179  1.00  0.00
ATOM    316  CD1 LEU A  40       8.886  -4.973  23.944  1.00  0.00
ATOM    317  CD2 LEU A  40       7.121  -6.784  24.145  1.00  0.00
               // Empty line - a point of a break
ATOM    318  N   GLU A  41       4.357  -6.093  21.040  1.00  0.00
ATOM    319  CA  GLU A  41       3.066  -6.778  21.082  1.00  0.00
ATOM    320  C   GLU A  41       2.967  -7.863  19.997  1.00  0.00
ATOM    321  O   GLU A  41       2.821  -9.046  20.315  1.00  0.00
ATOM    322  CB  GLU A  41       1.903  -5.775  20.992  1.00  0.00
ATOM    323  CG  GLU A  41       1.986  -4.741  22.132  1.00  0.00
ATOM    324  CD  GLU A  41       0.577  -4.464  22.689  1.00  0.00
ATOM    325  OE1 GLU A  41      -0.227  -5.435  22.661  1.00  0.00
ATOM    326  OE2 GLU A  41       0.371  -3.298  23.120  1.00  0.00
TER
```

Parameters:

Input	
Data	*.cov file, containing one or more sets of protein fragments
Output	
Result	Name of the output file with 3D protein structure in PDB format.
Options	
Number of Sets	Protein fragments sets number
Number of Steps	Number of cycles of optimisation (usually 100 - 1000).

CysRec

The program performs prediction of SS-bonding states of cysteines and locating of disulphide briges in proteins.

Methodology

Procedure: The sequence is processed in steps.

1. Secondary structure is predicted for a query sequence.
2. Amino acid fragment as well as fragment of secondary structure in ±10 positions interval of each cysteine is compared with such fragments of training sets using prepared log-odds matrix, and the maximal score is defined for each set.
3. Scores of comparisons with profiles (weight matrices) constructed on positive (bounded) and negative examples are calculated for a given fragment.
4. Value of linear discriminant function is calculated based on 4 the most significant amino acid properties.

5. The resulting score computed as a linear combination of five scores listed above is used for the recognition of SS-bonding states of cysteines.
6. A neural network calculates some scores for each possible pair of cisteines forming a 'Matrix of pair scores'.
7. A pattern of possible pairs of bounded cysteines is defined for maximum of sum of the scores of the matrix.

Input Format
Fasta formatted sequence divided by lines ≤ 80 positions in lengths is accepted.
Specially prepared alignment without gaps in the first sequence is accepted too.
Example of alignment:

```
 T0129
     5  182

 MLISHSDLNQQLKSAGIGFNATELHGFLSGLLCGGLKDQSWLPLLYQFSN
 ---SYSDFSQQLKTAGIALSAAELHGFLTGLICGGIHDQSWQPLLFQFTN
 -LPTYPSLALALSQQAVALTPAEMHGLISGMLCGGSKDNGWQTLVHDLTN
 ----YDEMNRFLNQQGAGLTPAEMHGLISGMICGGNNDSSWQPLLHDLTN
 ----YNEMNQYLNQQGTGLTPAEMHGLISGMICGGNDDSSWLPLLHDLTN

 DNHAYPTGLVQPVTELYEQISQTLSDVEGFTFELGLTEDENVFTQADSLS
 ENHAYPTALLQEVTQIQQHISKKLADIDGFDFELWLPENEDVFTRADALS
 EGVAFPQALSLPLQQLHEATQEALEN-EGFMFQLLIPEGEDVFDRADALS
 EGLAFGHELAQALRKMHAATSDALED-DGFLFQLYLPEDVSVFDRADALA
 EGMAFGHELAQALRKMHSATSDALQD-DGFLFQLYLPDDVSVFDRADALA

 DWANQFLLGIGLAQPELAKEKGEIGEAVDDLQDICQLGYDEDDNEEELAE
 EWTNHFLLGLGLAQPKLDKEKGDIGEAIDDLHDICQLGYDESDDKEELSE
 GWVNHFLLGLGMLQPKLAQVKDEVGEAIDDLRNIAQLGYDEDEDQEELAQ
 GWVNHFLLGLGVTQPKLDKVTGETGEAIDDLRNIAQLGYDESEDQEELEM
 GWVNHFLLGLGVTQPKLDKVTGETGEAIDDLRNIAQLGYDEDEDQEELEM

 ALEEIIEYVRTIAMLFYSHFNEGEIESKPVLH
 ALEEIIEYVRTLACLLFTHFQPQLPEQKPVLH
 SLEEVVEYVRVAAILCHIEFTQQKPTAKPTLH
 SLEEIIEYVRVAALLCHDTFTRQQPTAKPTLH
 SLEEIIEYVRVAALLCHDTFTHPQPTAKPTLH
```

Output Format
Query sequence
Positions of cysteines which are predicted to form disulfide bonds, matrix of pair scores results of SS-bonding states predictions, the most probable pattern of pairs.
Example of output:

```
>1AC5_
 length=483
LPSSEEYKVAYELLPGLSEVPDPSNIPQMHAGHIPLRSEDADEQDSSDLEYFFWKFTNNDSNGNVDRPLIIWLNGGPGCSS
MDGALVESGPFRVNSDGKLYLNEGSWISKGDLLFIDQPTGTGFSVEQNKDEGKIDKNKFDEDLEDVTKHFMDFLENYFKIF
PEDLTRKIILSGESYAGQYIPFFANAILNHNKFSKIDGDTYDLKALLIGNGWIDPNTQSLSYLPFAMEKKLIDESNPNFKH
LTNAHENCQNLINSASTDEAAHFSYQECENILNLLLSYTRESSQKGTADCLNMYNFNLKDSYPSCGMNWPKDISFVSKFFS
TPGVIDSLHLDSDKIDHWKECTNSVGTKLSNPISKPSIHLLPGLLESGIEIVLFNGDKDLICNNKGVLDTIDNLKWGGIKG
FSDDAVSFDWIHKSKSTDDSEEFSGYVKYDRNLTFVSVYNASHMVPFDKSLVSRGIVDIYSNDVMIIDNNGKNVMITT

7 cysteines are found in positions:   79  251  271  293  308  345  386


Matrix of pair scores
 POS:     79   251   271   293   308   345
   79: -999   -21    -4     8    18   143
  251:  -21  -999   155     7    -3   -12
  271:   -4   155  -999    13   -20   -15
  293:    8     7    13  -999   133    -8
  308:   18    -3   -20   133  -999    -7
  345:  143   -12   -15    -8    -7  -999
CYS     79 is SS-bounded               Score=  56.7
CYS    251 is SS-bounded               Score=  53.2
CYS    271 is SS-bounded               Score=  47.0
CYS    293 is SS-bounded               Score=  68.1
CYS    308 is SS-bounded               Score=  63.9
CYS    345 is SS-bounded               Score=  60.7
CYS    386 is not SS-bounded           Score= -70.7

The most probable pattern of pairs: 79-345, 251-271, 293-308,
```

Performance: 3000 positive and 3000 negative examples (i.e ± 10 fragments surrounding bounded and not bounded cysteines) were prepared from PDB sequences that were not participated in the training. An accuracy of SS-bonding states recognition by combined function on this control set was ~90%.

Parameters:

Input	
Sequence	Name of the input file.
Output	
Result	Name of the output file.

EnvFold

EnvFold is a program for search of homology of sequence with DB PDB sequences.

The Fold program searches for the homologues of a processed sequence in the PDB with use of files specially prepared by envbc program, which contain the following fields for each position:

- Amino acid in three letter code
- Area Buried
- Fraction Polar
- Secondary structure assignment

Keys for program run string:

1. Name of a file containing the processed sequence in FASTA format with size of not more than 1000 nucleotides and with strings' length of not more than 80 positions. As such a file, the specially prepared file of alignments of the processed sequence with other ones that does not contain gaps in test sequence can be used (see example for SSPAL program).
2. Name of a file containing the secondary structure of the processed sequence (see description for SSPAL or PSSF output files).
3. Name of the output file containing the results of comparison in the following format:
4. `T0234  165`
5.
6. `1VL7A Sc_b= 34906.0 Sc_lg= 1393.7 l2= 135`
7. `1G79A Sc_b=  3770.0 Sc_lg=  139.5 l2= 199`
8. `1G76A Sc_b=  3755.0 Sc_lg=  138.9 l2= 199`

The first string contains the name and length of tested sequence, the following ones - names of PDB sequences, common and relevant homology scores, and lengths of PDB sequences.

9. Aligning mode: 'f' - Global, 'l' - Local.
10. Name of the output file containing the alignment of the processed sequence with most homologous PDB sequence.
11. Name of a file containing the PDB sequence.
12. The path to DB files. The last symbol - '/'.

Fold

Program for search the homology of a processed sequence with sequences from PDB.

The Fold program searches for the homologues of a processed sequence in the PDB with use of files specially prepared by envbc program, which contain the following fields for each position:

- Amino acid in three letter code
- Area Buried
- Fraction Polar
- Secondary structure assignment

Program selects 100 cases with maximal similarity properties.

Keys for program run string:

1. Name of a file containing the processed sequence in FASTA format with size of not more than 1000 nucleotides and with strings' length of not more than 80 positions. As such a file, the specially prepared file of alignments of the processed sequence with other ones that does not contain gaps in test sequence can be used (see example for SSPAL program).
2. Name of a file containing the secondary structure of the processed sequence (see description for SSPAL or PSSF output files).
3. Name of the output file containing the results of comparison in the following format:

```
4.        T0234  165
5.
6.        1VL7A Sc_b= 34906.0 Sc_lg= 1393.7 l2= 135
7.        1G79A Sc_b=  3770.0 Sc_lg=  139.5 l2= 199
8.        1G76A Sc_b=  3755.0 Sc_lg=  138.9 l2= 199
```

The first string contains the name and length of tested sequence, the following ones - names of PDB sequences, common and relevant homology scores, and lengths of PDB sequences.

9. Aligning mode: 'f' - Global, 'l' - Local.
10. Name of the output file containing the alignment of the processed sequence with most homologous PDB sequence of the following type:

```
11. >T0283  112
12. 1ORJA Sc_b=  2385.0 Sc_lg=  104.5 l2= 126
13.          10        20        30        40        50        60
14.  aaaaaaaa     aaaaaaaaaaaaa  aaaaaaaaaaaaaa        aaaaaaaaaa
15.  MSFIEKMIGSLNDKREWKAMEARAKALPKEYHHAYKAIQKYMWTSGGPTDWQDTKRIFGG
16.  IECLERAIEIYDQVNELEKRKEFVENIDRVYD-IISALKSFLDHEKGKEIAKNLDTIYTI
17.  aaaaaaaaa       aaaaaaaaaaaaaaaa-aaaaaaa       aaaaaaaaaaaaa
18.          70        80        90       100
19.  aaaaaaaaaa         aaaaaaaaaaaaaaaaaaaaaaaaaa
20.  ILDLFEEGAAEGKKVTDLTGEDVAAFCDELMKDTKTWMDKYRTKLNDS
21.  ILNTLV-----KV---DKTKEELQKIL-EILKDLREAWEEVKKKVHHH
```

22. `aaaaaa-----   ---    aaaaaaaa-aaaaaaaaaaaaaaaaaa`
23. Name of a file containing the list of PDB sequences. Choosing a single id from the list, user can make an alignment of processed sequence exactly to chosen sequence independently of their similarity degree.
24. The path to DB files. The last symbol - '/'.

GetAtoms

The program GetAtoms allow to model spatial protein structure by homology. The model of the target protein structure is built using homologous template protein structure and pairwise sequence alignment of the template and target proteins. The program allows to:

- Calculate of the side chain atomic coordinates for the residues with known main-chain residues in the template protein structure;
- Model of the loop regions for which no main chain atomic coordinates in the template structure (insertions in the target protein in the pairwise sequence alignment);
- Model of main chain coordinates in the chain-break regions (deletions in the target sequence in the pair-wise sequence alignment).

The program allows to input alignment data in various formats. The model output can be performed in PDB or AMBER formats.
****

The approach is shown in the Fig.1.

Fig. 1. The approach of the GetAtoms program.

TSGS

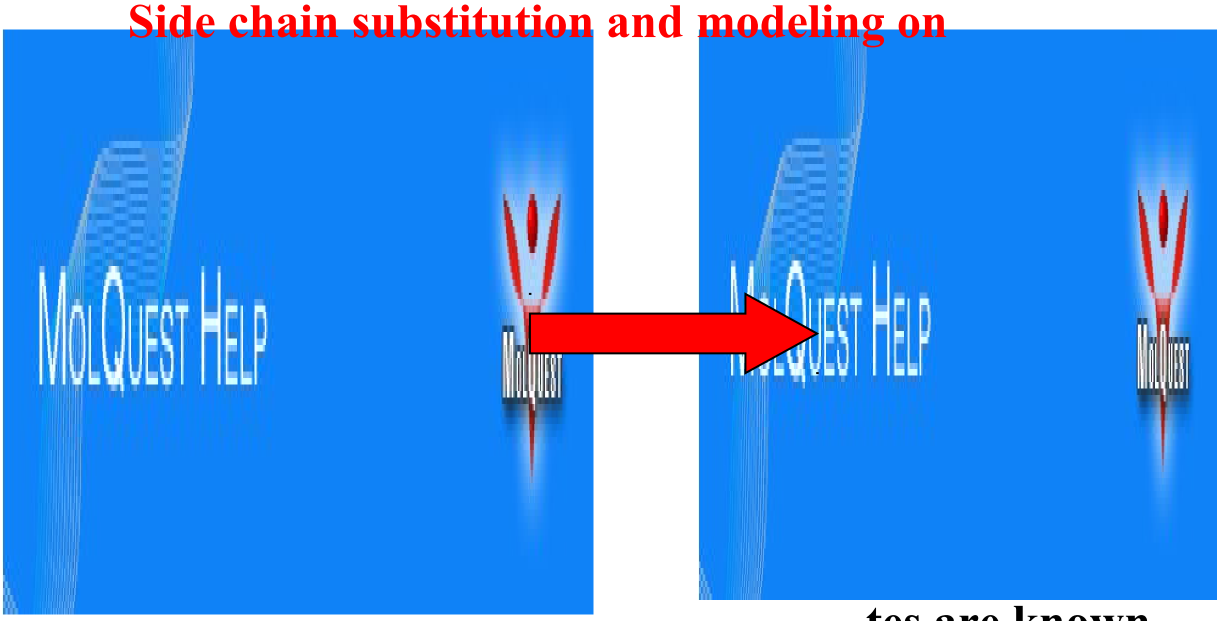
get alignment

...LDPGLPLPSRAHDGDAGVDLYSA...
...VGKEFPLPTY

Target sequence
Tem

loops modeled

Side chain substitution and modeling on



late sequence

tes are known

Template: backbone & side chain coordin

Target: backbone coordinates are from template, side chains modeled,

The program work in three stages.
First, the program makes side chain substitution in the template structure according to amino acid sequence in the target structure. Then rough preliminary side chain optimization is performed to remove steric clashes. The optimization is performed by Monte-Carlo algorithm and is as follows. Initially the side chain is placed in most frequent rotameric state. Then program searches for the side chains that form clashes and try to change their conformation randomly. If the sterical energy is lower than the energy at the previous step, new configuration is accepted. If not, the energy change dE is calculated and the value of exp(-dE/*Temperature*) is compared to the random number *rand* in the range [0,1]. If *rand* value is lower, such conformation is accepted. The *Temperature* specifies the temperature for MC algorithm of side chain conformation optimization, the lower the temperature, the faster is the convergence to the nearest local minima. Higher temperature allows overcoming local minima but needing more time for search. This procedure is repeated user-defined maximal number of MC steps (for the preliminary optimization the number of 50-100 for this parameter is recommended). Sometimes the side chain rotamer configuration can be trapped in the state with high sterical energy, to overcome this, it is useful to make restart from random configuration of rotamers to new optimal configuration if optimization is not successful in 100 steps. The restart is controlled by MC process restart option.

Second step performs main chain reconstruction in the insertion and deletion regions of the template-target superposition (Fig. 2).

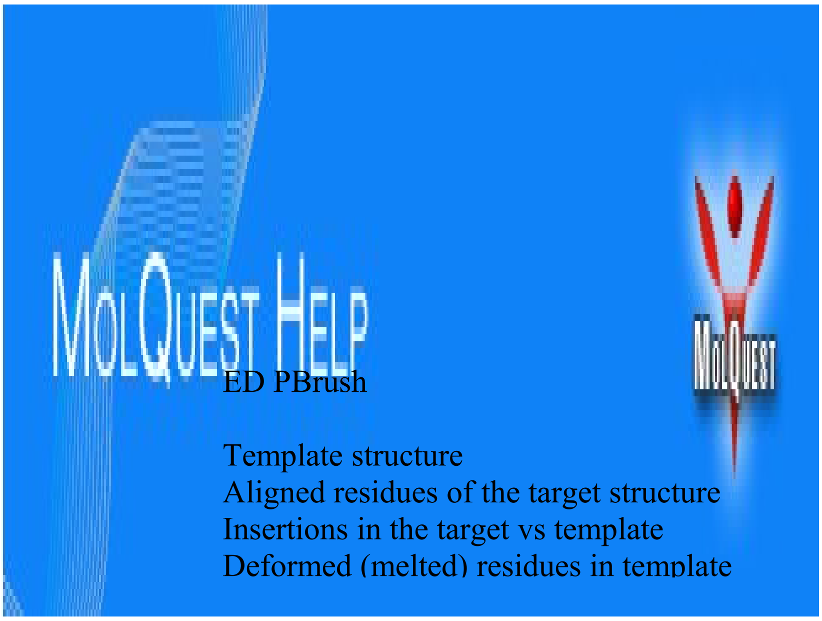

MolQuest Help
MolQuest
ED PBrush
Template structure
Aligned residues of the target structure
Insertions in the target vs template
Deformed (melted) residues in template

Fig 2. The insertion modeling approach.

During insertion modeling the program try to generate many loop main chain conformation in attempt to “close” the space gap between the C-terminus of the loop and N-terminus of the residue immediately following after the insertion. These conformations are generated by Monte Carlo procedure and controlled by temperature and maximal number of iteration steps as described previously. Conformations that have the distance between loop C-termini modeled N-atom and the true anchor N-atom less then user-defined threshold (C-ter attachment criterion) then screened for the conformation that have minimal sterical energy of interaction with the other part of the protein. Note, that the two template residues immediately at the place of the insertion are “melted” (actually they are added to the loop) to make local distortion in the template to allow loop to be inserted.

The same procedure is implemented for deletions modeling (Fig 3).

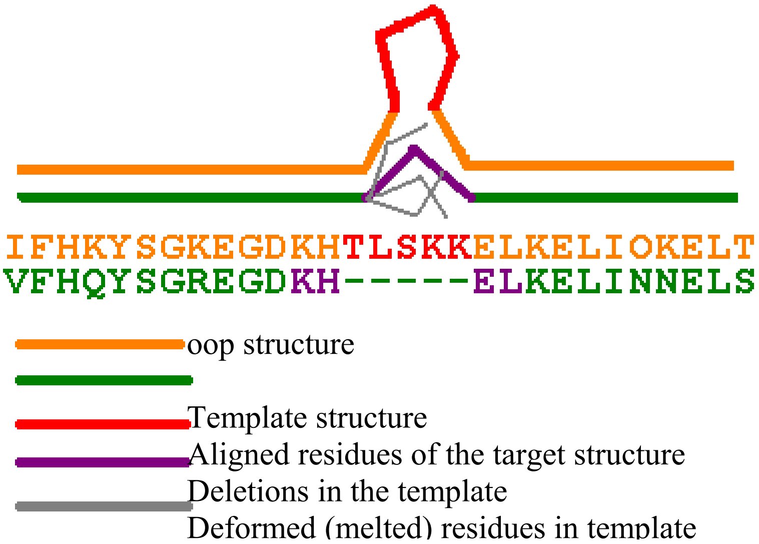

IFHKYSGKEGDKHTLSKKELKELIOKELT
VFHQYSGREGDKH-----ELKELINNELS
oop structure
Template structure
Aligned residues of the target structure
Deletions in the template
Deformed (melted) residues in template

Fig 3. The deletion modeling approach.

In this case two residues from both termini of the deletion are “melted” (actually they are formed a loop from 4 residues), that is build by previous algorithm.

After the insertion and deletion modeling the final optimization step is performed for side chain conformations only. The algorithm is the same as for the first step, but it is recommended to make the number of optimization steps larger (200-400).

The user can also control additional input and output parameters.
Alignment format: format of the alignment file. Several options are possible. "LOCAL", the output format of the Softberry FOLD program; "FASTA", FASTA-format; "SIMPLE", format with only sequences in the data (no sequence names); "CE", alignment format from the CE structural alignment program. First sequence is the target, second sequence is template. Columns of alignment containing only gaps in both sequences are ignored.
Adding Hydrogen atomsHAtoms {ON,OFF}: the coordinates of the hydrogen atoms will be added to heavy atoms in the modeled structure.
StatusFile : the name of the file for calculation status output
SaveFormat : output format, PDB, the PDB format; AMBER the structure ormat that can be read by AMBER program.
BumpedList : filename with the list of atomic clashes that was not resolved by GetAtoms program.

The output file contains some information about the optimization parameters and initial and final energy of the protein structure.

GetAtoms output:

```
HEADER    OXYGEN TRANSPORT                        07-MAR-84   4HHB
REMARK  50
REMARK  50 GETATOMS [ver=0.9.0.0; date=20020312]
REMARK  50 Modelled from template structure provided by user.
REMARK  50 Calculation parameters:
REMARK  50   Simulated Annealing Temperature=2.000000
REMARK  50   Simulated Annealing Maximal number of steps=100
REMARK  50   Simulated Annealing steps done=-1073216864
REMARK  50   Add Hydrogen Atoms=OFF
REMARK  50 Final score data:
REMARK  50   VDW_Score=1.089206e-19
REMARK  50   Steric_Score=2.652495e-315
REMARK  50   Bump_Score=0.000000e+00
ATOM      1  N   VAL     1       9.223 -20.614   1.365
ATOM      2  CA  VAL     1       8.694 -20.026  -0.123
ATOM      3  C   VAL     1       9.668 -21.068  -1.645
ATOM      4  O   VAL     1       9.370 -22.612  -0.994
ATOM      5  CB  VAL     1       8.948 -18.511  -0.251
ATOM      6  CG1 VAL     1       8.554 -18.010  -1.636
ATOM      7  CG2 VAL     1       8.176 -17.751   0.822
ATOM      8  N   LEU     2       9.270 -20.650  -2.180
ATOM      9  CA  LEU     2      10.245 -21.378  -3.143
ATOM     10  C   LEU     2      11.419 -20.331  -4.099
ATOM     11  O   LEU     2      11.252 -19.250  -5.024
ATOM     12  CB  LEU     2       9.461 -22.198  -4.174
ATOM     13  CG  LEU     2       8.651 -23.375  -3.627
ATOM     14  CD1 LEU     2       7.843 -24.024  -4.741
ATOM     15  CD2 LEU     2       9.576 -24.392  -2.976
ATOM     16  N   SER     3      12.365 -20.722  -3.649
ATOM     17  CA  SER     3      13.611 -20.183  -4.477
ATOM     18  C   SER     3      14.557 -21.356  -5.125
ATOM     19  O   SER     3      14.340 -22.536  -4.780
ATOM     20  CB  SER     3      14.497 -19.299  -3.595
ATOM     21  OG  SER     3      15.076 -20.068  -2.554
```

or WITH H-atoms:

```
REMARK  50   Add Hydrogen Atoms=ON
REMARK  50 Final score data:
REMARK  50   VDW_Score=1.089206e-19
REMARK  50   Steric_Score=2.652495e-315
REMARK  50   Bump_Score=0.000000e+00
ATOM      1  N   VAL     1       9.223 -20.614   1.365
ATOM      2  CA  VAL     1       8.694 -20.026  -0.123
ATOM      3  C   VAL     1       9.668 -21.068  -1.645
ATOM      4  O   VAL     1       9.370 -22.612  -0.994
ATOM      5  CB  VAL     1       8.948 -18.511  -0.251
ATOM      6  CG1 VAL     1       8.554 -18.010  -1.636
ATOM      7  CG2 VAL     1       8.176 -17.751   0.822
ATOM      8 1H   VAL     1      10.102 -20.497   1.435
ATOM      9 2H   VAL     1       8.812 -20.175   2.021
ATOM     10 3H   VAL     1       9.034 -21.482   1.426
ATOM     11  HA  VAL     1       9.166 -20.592  -0.926
ATOM     12  HB  VAL     1      10.006 -18.305  -0.091
ATOM     13 1HG1 VAL     1       9.071 -17.073  -1.845
ATOM     14 2HG1 VAL     1       8.833 -18.752  -2.384
ATOM     15 3HG1 VAL     1       7.477 -17.846  -1.671
ATOM     16 1HG2 VAL     1       7.168 -17.540   0.463
ATOM     17 2HG2 VAL     1       8.120 -18.356   1.727
ATOM     18 3HG2 VAL     1       8.686 -16.814   1.043
ATOM     19  N   LEU     2       9.270 -20.650  -2.180
ATOM     20  CA  LEU     2      10.245 -21.378  -3.143
ATOM     21  C   LEU     2      11.419 -20.331  -4.099
ATOM     22  O   LEU     2      11.252 -19.250  -5.024
ATOM     23  CB  LEU     2       9.461 -22.198  -4.174
ATOM     24  CG  LEU     2       8.651 -23.375  -3.627
ATOM     25  CD1 LEU     2       7.843 -24.024  -4.741
ATOM     26  CD2 LEU     2       9.576 -24.392  -2.976
ATOM     27  H   LEU     2       8.525 -20.036  -1.884
ATOM     28  HA  LEU     2      10.867 -22.070  -2.576
ATOM     29 1HB  LEU     2       8.746 -21.553  -4.685
ATOM     30 2HB  LEU     2      10.152 -22.623  -4.903
ATOM     31  HG  LEU     2       7.969 -23.019  -2.854
ATOM     32 1HD1 LEU     2       7.705 -23.310  -5.553
ATOM     33 2HD1 LEU     2       8.376 -24.899  -5.114
ATOM     34 3HD1 LEU     2       6.870 -24.328  -4.356
ATOM     35 1HG2 LEU     2       9.162 -24.699  -2.016
ATOM     36 2HG2 LEU     2       9.673 -25.263  -3.625
ATOM     37 3HG2 LEU     2      10.558 -23.944  -2.822
```

Parameters:

Input	
Template structure file	Data with template protein structure in PDB format
Template chain	This parameter specifies chain index in template structure to use as model. It shoud contain 1-letter symbol code or '_' symbol for chain without index (' ') in PDB file.
Alignment file	Data with target-template sequence alignment. Target is first sequence in alignment, template is the second.
Alignment format	Specifies alignment file format: **Simple alignment format** **FASTA format** **Local format output by FOLD program** **Format of alignment by CE program**
Output	
Result	Output file.
Format	Specifies format for output structure file: **PDB format output** **AMBER format output**
Status file	The calculation status file.
Options	
Optimization temperature	Specifies temperature for MC algorithm of side chain conformation optimization.
Adding hydrogen atoms	Specifies the addition of hydrogen atoms to final protein model structure.
Multiple chain processing	Specifies the accounting for additional protein chains in template structure. If 'false' only chain specified in "Template chain" parameter left. If 'true', other chains are left in final structure.

MolDyn

Preference

The Program **MolDyn** is designed to perform multiple tasks with protein structure:

1) restoration of missing coordinates of heavy atoms of side chains;

2) restoration of missing coordinates of all hydrogen atoms;

3) optimization of a protein structure via local energy optimization in an implicit/explicit water solvent;

4) optimization of a protein structure via MD simulation in water solvent;

5) optimization and folding of a protein via a user defined simulated annealing protocol coupled with force field variation.

6) optimization of a user defined flexible protein segments with user defined restraints

7) simulation of the molecular dynamical trajectory of atomic coordinates and potential energy for statistical analysis.

I. Input and Compilation

1. RUN the program

RUN program by the command

```
   ../$MDYN07HOME/mDynQ07 -i inProtcol  -c inPDB [-mdR mdRestXYZVin]
             [-mv moveRes]
             [-r1 inRestrainA1 ] [-r2 inRestrainA2] [-rB rigBodyFile]
             [-sa saProtocol] [-mn molName]  [-mdX mdFinalPDB] -o runOutFile
             [-er errorFile]
```

in parenthesis [ ] are uxilarry files. The auxilary files will be used by program if the main command file
defines respective task.

Command line DESCRIPTION:

-i inProtcol   : file MdynPar.inp   defines protocol for the mDyn particular Run

-c inPDB   : file of the initial molecular structure as molec.pdb file in the PDB format

```
-mdR mdRestXYZVin   : XYZ+Velocity file to REstart MD from the last snapshot
file XYZV , see exaple t5
                       1arb.mdXYZVfin0001.pdb it is USED with $mdRestart
keyword in command file
                    inProtcol
                      NOTE!  the initial XYZ will be taken from mdRestXYZVin
file !
                          the PDB file inPDB is not USED with the key -mdR
```

```
-r1 inRestrainA1   : file defines positional restraints for atoms of
                     the  molecule
-r2 inRestrainA2   : file defines atom-atom distance restraints
-rB rigBodyFile    : file defines rigid body segments of the main chain of
                     protein
-mv moveRes        : file defines List of moving Residues
-sa saProtocol     : file defines simulated annealing protocol
-mn molName        : character set defining molecula name.
                     molName. will be attached to RESULT files
-o runOutFile      : run output file
-mdX mdFinalPDB    : final PDB file of the Energy/MD optimization
```

Current status of program run is printed on the standart output device (consol) or
can be redirected to user defined file or can be defined in the argument line:
-er errorFile   : error message file : they are dublicated in the runOutFile
#
if file name definition in the argument line is missing for a file
than the default name is used for this file

NOTE! if the command line does not include a key -X , while the command file defines task which need data file coupled with -X keyword, than program try to find default (standart) name data file in the current directory.
Default names:

```
#
inProtcol    = ./MdynPar.inp
inPDB        = ./molec.pdb
mdRestXYZVfile = ./mdXYZVin.pdb
moveRes      = ./moveRes.inp
```

```
inRestrainA1  = ./restrAt1.inp'
inRestrainA2  = ./restrAt2.inp'
rigBodyFile = ./rigBody.inp
saProtocol  = ./SAprotocol.inp
molName     = space
runOutFile  = ./mDynSB.out
errorFile   = ./mDynSB.err
mdFinalPDB  = ./molMdFin.pdb
#
```

2. Input file and keyword description

```
inProtcol   = ./MdynPar.inp

The nain command file consist of lines with command keyword.
Keyword start with $ sign in the first position of line
One Keyword in line

#example of MdynPar.inp file and keyword description
# MdynPar.inp
$OUTfull                            ! full extended output of program run

#Initial PDB data quality
$Hread                              ! read INPUT pdb file with Hydrogens
                                    ! by default OUTshort option is ON
# DEfinition of OPtimized segments of protein:
$fullProtMD                         ! full molecule is flexible
$MovingRes                          ! defines List of opimized segments

#FORCE FIELD MODIFICATIONS:
#
$shake=2                        ! all valence bobds are fixed by shake method

$zeroRot                   ! exclude translation and rotation of the molecule
                                          as rigid body

$hBond128 = 2.0                      ! scaling coeff for H-bonds
                                     ! default=1.0 it is standart force field

$harmAt1PosRst=0.25              !invoke restraintsA1 type =
                                 positional harmonic restraints for
                                 atom position
                                 with harmConst (kcal/A^2).
                                 program need a special file -r1 restrA1File
                                 which defines restrained segments of protein
                                 (see additional description)

$distRestrA2                    !invoke restraintsA2 type atom-atom distances
                                  for user defined pairs of atoms in the file
                               -r2 restrA2File (see additional description)


$rigBody                !invoke optimization with frozen internal structure of
                      protein main chain for user defined segments of sequence
                   need file -rB rigidBodySegment (see additional description)

$compactForce = 0.5                 ! invoke additional compactization forces
                                    ! to accelerate protein folding
#
$aSoftCore = 0.5        !invoke SOFTNES for the van der waals
                                  atom-atom potential
                        ! at the small (contact) atom-atom distances
                        ! Use of the softCore VDW potential helps to optimize
                        ! BAD molecular structures with many spartial
```

```
                           atom-atom clashes
                        ! values range  0 - 1 from very Soft to standard VDW
#SOLVATION MODEL
$SolvMod = GShell
#
#
# OPIMIZATION PROTOCOL:
$engCalc                          ! do energy calculation
$engOptim                         ! do energy optimization by local Optimizer
$nOptStep=11                      !max N optim steps
#
#PROTOCOL for Molecular Dynamics:
$doMDyn                                ! do MolDynamics
$MDSA                                  !do MolecularDynamis SimAnnealing
                                 needs SAprotocolFile -sa saProtocol File,
                                        see additional description
#
#PROTOCOL of MD equilibration:
#
$initMDTemp=50.00                      !initial Temperature to start MolDyn
$bathMDTemp=50.00            !thermostat temperature of thermostat i.e. target
temperature
$runMDnstep=2000           !number of time-steps for MD simulation
$mdTimeStep=0.002
#
$NTV=1                                ! MD ensemble definition
#
#
# MD Trajectory writing:
$nwtra=500
$WRpdb                        ! write snarshort structures in the PDB format
                          ! default WRpdbq OPTion is ON : extended PDB format
                                         ! PDB + Qatom
#
END
#
NOTE that parameter file formatted, i.e. $ sign should be  the firs character
of the line
----------------------------------------------------------------------------
KEYWORD LIST:
        keyw = 'OUTfull'
        keyw = 'WRpdb'
        keyw = 'Hread'
        keyw = 'fullProtMD'
        keyw = 'MovingRes
        keyw = 'MDSA'
        keyw = 'SolvMod'
        keyw = 'zeroRot'
        keyw = 'hBond128'
        keyw = 'harmAt1PosRst'
        keyw = 'distRestrA2'
        keyw = 'compactForce'
        keyw = 'shake'
        keyw = 'engCalc'
        keyw = 'engOptim'
        keyw = 'nOptStep'
        keyw = 'aSoftCore'
        keyw = 'initMDTemp'
        keyw = 'bathMDTemp'
        keyw = 'mdTimeStep'
        keyw = 'runMDnstep'
        keyw = 'doMDyn'
        keyw = 'mdRestart'
        keyw = 'NTV'
        keyw = 'nwtra'
------------------------------------------------------------------------------
KEYWORD DESCRIPTION:

#OUTPUT DETAILES:
$OUTfull                                    ! full extended output of program
run
                                         ! by default OUTshort option is ON
#
# INPUT PDB FILE DETAILES:
$Hread      ! defines that all Hydrogens will be read from input molecule
structure -c inPDB   file
            otherwise the ALL HYDrogens will be restored by the program, i.e.
               all H atoms will be deleted and added according to molecular
topology for RESidues.
            Using Library in the ./dat/h_add.dat
NOTE! it is recommended start to works with a new protein without option
$Hread even if the PDB
file has all hydrogen atoms, because the hydrogen atom names for protein side
chains
have multiple definition in the PDB data base.
It is better if mDyn  program will add all hydrogens to the heavy atoms.

#DEFINITION OF OPTIMIZED RESIDUES:

$fullProtMD                              !defines FULL (i.e. ALL atoms) of the
USER molecule
                                               will be free to move in energy
relaxation or molDyn

$MovingRes      ! logical keyWord defines that only a defined set
                  of RESidue are free
                 this keyWord is coupled with file -mv moveRes in
                 the argument line to start
                 the program
                 default name for moveRes file is ./moveRes.inp

#EXAMPLE of ./moveRes.inp
#1arb
aaaaaaIIIIiiii
#
MOVRES   1  10       !line defines first and last residue
                       of moving segments integers devided by space
MOVRES  45  76
MOVRES 115 260
end                  !end or END should be last line if the file
************

#FORCE FIELD DEFINITION:

$hBond128 = 2.0                          ! scaling coeff for H-bonds

$aSoftCore = 0.5                   !invoke van der waals atom-atom potential
                                    with modified repulsion
                                   ! SoftCore at the small (contact)
                                     atom-atom distances
                        ! SoftCore modification is used for
                          energyOPtimization
                          and MD equilibration stages.
                        ! Use of the softCore VDW potential helps to optimize
                        ! BAD structures with many starical atom-atom clashes
                        ! values range  0 - 1 from very Soft to standart VDW

$harmAt1PosRst=0.25  ! digital keyWord define RESidue segments with 1
```

```
                       atom position harmonic restrants.
                       0.25 = harmonic restrain Constant K
                       restrEnergy = 0.5*K(r - r0)**2,
                       the reference position r0 = initialXYZinput.pdb
                       - positions from
                       the initial INPut PDB file which defines
                       the INItial structure of molecule
                       this keyWord is coupled with file -r1 inRestrainA1 of
                       the argument line to start the program mdyn
                       default name for inRestrain file is ./restrAt1.inp

#EXAMPLE  of inRestrainA1 file:
#harmonically restrained RESidue segments
#xxxxxIIIIiiiiaaAAA
#(6x,2i4,a40)
RESTA1   1  63  PBB   ! line starts from keyWord RESTAT numbers=first/last
                        residue of segment
                     ! PBB (only protein backbone atoms are restrained, i.e.
                        side chains are free)
RESTA1  78 120  ALL   ! ALL (all atoms are restrained)
                       ! integers and words are devided by space
end
# ---------------------------------------------------
$distRestrA2         ! defines optimization/MD with atom-atom dist RestrainA2
                     ! needs file  [-r2 inRestrainA2] in command line
-r2 inRestrainA2 : default name : restrAt2.inp
#
EXAMPLE of inRestrainA2 file:
#harmonically restrained Atom-Atom distances
#xxxxxx
#keyword atom1       atom2       distA HarmConst(kcal/mol*A^2)
RESTA2   ND2  ASN 222 : OG1 THR 219 = 7.0   1.5
RESTA2   O    GLY 170 : OG1 THR 219 = 8.0   2.5
RESTA2   OH   TYR 109 : OG1 THR 111 = 7.5   3.0
END
#----------------------------------------------------
$rigBody                   !defines optimizatiom/MD considering some segments
                            of the main chain
                            ! as a rigid body.
                            ! The List of rigid  segments of the main chain
                              is user defined.
                            ! Each segment will keep rigid internal structure
                              of the protein main chain,
                               ! has rotatational and translational degrees of
freedom.
                                  ! The side chains of the rigid segments are
flexible.
#Needs file rigidBody.inp
#EXAMPLE of rigidBody.inp file:
#
RIGB01  11  16       !line defines first and last resudue of moving segments
                       integers devided by space
RIGB02  47  59
RIGB03  77  99
end                  !end or END should be last line if the file
# - - - - - - - - - - - - - - - - - - - - - - - - -
$compactForce = 0.25           ! define additional compactization forces for
protein atoms
                            ! Recomended forceParameter = 0.1 - 1.0
# --------------------------------------------------

$shake=2    ! invoke shake subroutine to keep bonds fixed. =1 -bonds with
              Hydrogen, =2 all bonds
```

```
----------------------------------------------------
#Defining of the SOLVation model:
there are 4 variants  of Implicit models
          1 variant of Explicit model
#:
$SolvMod = GShell           ! implicit  Gaussian Shell solvation model
$SolvMod  =  GShell  +  WBrg      !  implicit   Gaussian  Shell  solvation  model  +
WaterBridges between polar atoms
                                    !  WaterBridges  descride  solvent  mediated
interactions trough stong bound water
                         ! molecules via implicit model of water bridges

$SolvMod  =  GBorn                  !  implicit  Generalized  Born  model  +  SAS
HydroPhobic solvation
$SolvMod  =  GBorn  +  WBrg         !  implicit  Generalized  Born  model  +  SAS
HydroPhobic solvation + WaterBridges

$Solv = ExWshell 4.5 [A] ! explicit water shell of 4.5 Angst around protein;
                         ! recomended thikness 3.0 - 6.0 A
---------------------------------------------------
$mdRestart                !    restart    molDynamics    from    a    snapshot
[molName.]mdXYZVfin000N.pdb
                 the file [molName.]mdXYZVfin000N.pdb should be copied to the
file mdyn Restart file
                mdXYZVin.pdb

$doMDyn       ! do molecular dynamics
$MDSA         ! do Molecular Dynamical Simulated Annealing
               ! coupled with file -sa SAprotocol which define protocol of the
simulated annealing

#EXAMPLE of Aprotocol.inp  file
#SA protocol
#nSAstep 2
#(f10.1,1x,f8.1,1x,3(f6.1,1x)
#      nMDstep    tempTg   SCvdW wfHb128BB wfhB128BS
SAPROT 100000      500.0     0.8    1.0     1.0        !line starts with
                                                        keyword SAPROT
SAPROT 100000      100.0     1.0    1.0     1.0
END
#
   nMDstep - number of md timeStep
   tempTg  - target temperature in K, this temperature will be
   reach during ntimeMX steps
   SCvdW   - parameter 0 - 1 to define softness of the van der waals
              potential. Soft potential
             modifies Potential Energy Surface and decrease  barriers of
             conformational transitions
   wfHb128BB,
   wfhB128BS - (1 - 0) scaling factors for BackBone-BackBone and
                BackBone-SideChain Hydrogen Bond energy
#---------------------------------------------------
#
# OPIMIZATION PROTOCOL:
$engCalc                               ! do energy calculation
$engOptim                                   !  do  energy  optimization  by  local
Optimizer
$nOptStep=1                            !max N optim steps
#
#PROTOCOL for Molecular Dynamics:
$doMDyn                                ! do MolDynamics
$MDSA                                  !do MolecularDynamis SimAnnealing
                                          needs SAprotocolFile -sa saProtocol
File,
#MD EQUILIBRATION:
$initMDTemp=50.00                       !defines initial temperature to start MD
                                        ! recommended low temperature < 50K
                                         ! temperature can be steadelly increased
to the 300K and higher
                                        ! USING $MDSA option
$bathMDTemp=50.00                                  !  bath  temperature  in  the  MD
equilibration run
$runMDnstep=2000                              ! number of MD time steps   in the
equilibration run
$mdTimeStep=0.002                       ! value of the MD time step in ps,
                                        ! recomended 0.001 - 0.002
$NTV=1                                  ! ansemble NTV=0/1
                                        ! =1 md run with constant T
#MD TRAJECTORY WRITING
$nwtra=500                                    ! structure XYZ  (snapshot)  will be
written
                                        !as a series of molMdResXXXX.pdb files

$WRpdb                                      ! write snapshort structures in the
                                              PDB format
                                            ! default is WRpdbq OPTion is ON :
                                              extended PDB format
                                            ! PDB + Qatom column
#* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * *
#
-c inPDB   file  - standart pdb file

#EXAMPLE of inPDB   file:
*****************************************************************************
***********
NOTE! it is recommended to start to work with a new protein without option
$Hread even if the PDB
file has all hydrogen atoms, because the hydrogen atom names for protein side
chains
have multiple definition in the PDB data. It is better if mDyn  program will
add all hydrogens
to the heavy atoms.
*****************************************************************************
**************
REMARK: PDB:
ATOM      1  N   GLY A   1      11.726 -10.369  10.598
ATOM      2  H1  GLY A   1      11.921 -11.015   9.807
ATOM      3  H2  GLY A   1      12.518 -10.395  11.271
ATOM      4  H3  GLY A   1      10.852 -10.663  11.079
ATOM      5  CA  GLY A   1      11.567  -9.015  10.090
ATOM      6  HA2 GLY A   1      10.772  -8.977   9.420
ATOM      7  HA3 GLY A   1      12.439  -8.710   9.612
ATOM      8  C   GLY A   1      11.280  -8.099  11.303
ATOM      9  O   GLY A   1      11.256  -8.584  12.493
ATOM     10  N   VAL A   2      11.060  -6.876  11.020
ATOM     11  H   VAL A   2      11.066  -6.574  10.025
etc.
TER                   ! CHAIN TERmination
ATOM   1302  N   GLY A  94      10.957 -15.678  12.832
ATOM   1303  H   GLY A  94      10.735 -14.663  12.877
ATOM   1303  H   GLY A  94      10.735 -14.663  12.877
ATOM   1304  CA  GLY A  94      10.193 -16.559  11.950
ATOM   1305  HA2 GLY A  94       9.428 -16.004  11.516
ATOM   1306  HA3 GLY A  94       9.784 -17.323  12.525
ATOM   1307  C   GLY A  94      11.016 -17.184  10.843
...
```

```
etc.
TER                 ! CHAIN TERmination
END                 ! file  END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
#
# PDB mDyn trajectory file description:
#
```

Program mDyn generate a series of snapshot files, e.g., 1arb.molMdRes0nnn.pdb (test/t4)
the molMdResXXXX.pdb file (see example) contains all atomic coordinates and additional information
in the REMARK: lines

```
####
REMARK: Md result : MdTime(ps):    2.4940
REMARK: $nstep:      1247
REMARK: $nRecPDB:       5
REMARK: RMSD(x0):     0.43   <- RMSD all atom
REMARK: badBond: n,erAv(A)  :     0  0.000    <- number and error Average for bond length in Angstrem
REMARK: badAng : n,erAv(grd):     8   9.42    <- number and error Average for bond angles in grad
# ENERGY TERMS for the given structure
REMARK: $ENERGY:     :Kcal
REMARK: eVbondDef:     100.89315      <-bond deformation energy
REMARK: eVangDef :     441.63705      <-angle deformation energy
REMARK:  eImpDef   :       35.68147       <-Improper torsion agle [planarity] energy
REMARK: eTorsDef :     691.25769      <-torsion potentioal energy
REMARK: engVDWR1 :   -1031.16211      <- van der waals energy for cutoff R1=8 A
REMARK: ehBHxY128:    -608.70599      <- H-bondinds energy
REMARK: engCOULR1:    -816.25323      <- COULOMBIC for distances < cutoff R1
REMARK: engCOULR2:      -4.47208     <- COULOMBIC for distances Rij,  R1< rij
```

3.TYPICAL EXAMPLES DESCRIPTIONS

# TEST#1:

# directory ./t1 shows example
to restore Hydrogen atoms whith geometry optimization and molDyn equilibration

The input PDB file does not have Hydrogen atoms

Job1:
1) add Hyrdogen atoms
2) make energy optimization
3) make molDynamics equilibration

command parameter file:
t1_MdynPar.inp :

```
#234567890123456789012345678901234567890!comment
#
$fullProtMD                             !all protein atom are moving i.e. are optimized
$SolvMod = GShell                      !USE Gaussian Solvation Shell Model
```

```
$engCalc                                ! do energy calculation
$engOptim                                 ! do energy Optimization for moving
atoms
$nOptStep=10                            ! max N optim steps
$doMDyn                                 ! do MolDynamics
$initMDTemp=10.00                       ! initial Temperature in Kelwin
$bathMDTemp=100.0                       ! thermal bath final Temperature
$runMDnstep=2000                        ! do 2000 moldyn steps
$mdTimeStep=0.001                       ! length of mdstep in ps
$nwtra=200                                ! write snapshot PDB files each 200
md steps
#END
```

Run the test1:

```
>  $MDYN011HOME/mDynQ011 -i t1_MdynPar.inp -c 1arb.0.pdb -mn 1arb -o t1.out
>
#NOTE!  this  command file t1_MdynPar.inp does not include $Hread keyword
therefore the program mdynQ09 will add XYZ of all Hydrogen atoms.
ALSO if some heavy atom of side chains are missing in the initial PDB file,
the program mdynQ09 will calculate coordinates of the side chain heavy atoms.
SEE test2.
#
TEST 1: console print out:
Status: 1 run mDynQ011 ...
 Status: 2 run mDynQ011 ...         Finish addHeavyAtom ...
 Status: 3 run mDynQ011 ...         file :molAddHvyAt.pdb
  is written ...
 Status: 4 run mDynQ011 ...         Finish add_Hatoms ...
 Status: 5 run mDynQ011 ...         made molec topology ...
 Status: 6 run mDynQ011 ...         made [solvated] molec topology ..
 Status: 7 run mDynQ011 ...         init ForceField parameters ..
 Status: 8 run mDynQ011 ...         init Gauss Shell solvation model ..
 Status: 9 run mDynQ011 ...         mdSnap :  1 is wrote ...
 Status: 10 run mDynQ011 ...         initialXYZ energy calculation done ...
 Status: 11 run mDynQ011 ...         start energyOptimization ...
 Status: 12 run mDynQ011 ...         mdSnap :  1 is wrote ...
 Status: 13 run mDynQ011 ...         energyOpimization is done ...
 Status: 16 run mDynQ011 ...         start molDyn run ...
 Status: 17 run mDynQ011 ...         mdSnap :  1 is wrote ...
 Status: 18 run mDynQ011 ...         mdSnap :  2 is wrote ..
 Status: 18 run mDynQ011 ...         mdSnap :  2 is wrote ...
 Status: 19 run mDynQ011 ...         mdSnap :  3 is wrote ...
 Status: 20 run mDynQ011 ...         mdSnap :  4 is wrote ...
 Status: 21 run mDynQ011 ...         mdSnap :  5 is wrote ...
 Status: 22 run mDynQ011 ...         mdSnap :  5 is wrote ...
 Status: 23 run mDynQ011 ...        mdRunCall Finish: StepDone=ntimeS: 1004
 Status: 24 run mDynQ011 ...         eqvilibration mDyn is done ...
 Status: 25 run mDynQ011 ...         successful finish mDynSB program ...
#
*******************************************************
#
# TEST#2:  initial PDB file has missing side chain atoms for some residues,
           no all Hydrogen atoms
#
test2
restore missing side chain atoms for
VAL2, ASN7, ILE8, ARG18  which have missing side chain atoms in the initial
pdb data file,
add Hydrogens,
do energy optimization and moldyn equilibration

In the Test #2
1) program AUTOMATICALLY adds all missing (in the initial pdb data file) side
chain atoms
```

2) add all Hyrdogen atoms
3) makes energy optimization
4) makes molDynamics equilibration

```
#files to run test2:
#t2_MdynPar.inp
#234567890123456789012345678901234567890!comment
$fullProtMD                             !all protein atom are optimized
$SolvMod = GShell                       !USE Gaussian Solvation Shell Model
$engCalc                                ! do energy calculation
$engOptim                                    ! do energy Optimization for all
atoms
$nOptStep=3                             ! max N optim steps
$doMDyn                                 ! do MolDynamics
$initMDTemp=10.00                       ! initial Temperature in Kelwin
$bathMDTemp=100.0                       ! thermal bath final Temperature
$runMDnstep=2000                        ! do 2000 moldyn steps
$mdTimeStep=0.001                       ! length of mdstep in ps
$nwtra=200                                ! write snapshot PDB files each 200
md steps
#END
~
```

run test2 by command

```
> $MDYN011HOME/mDynQ011 -i t2_MdynPar.inp -c 1arb.0.noHeavyAt.pdb -o t2.out
```

# program prints on console status of calculations:

```
tatus: 1 run mDynQ011 ...
 Status: 2 run mDynQ011 ...         Finish addHeavyAtom ...
 Status: 3 run mDynQ011 ...         file :molAddHvyAt.pdb
  is written ...
 Status: 4 run mDynQ011 ...         Finish add_Hatoms ...
 Status: 5 run mDynQ011 ...         made molec topology ...
 Status: 6 run mDynQ011 ...         made [solvated] molec topology ..
 Status: 7 run mDynQ011 ...         init ForceField parameters ..
 Status: 8 run mDynQ011 ...         init Gauss Shell solvation model ..
 Status: 9 run mDynQ011 ...         mdSnap :  1 is wrote ...
 Status: 10 run mDynQ011 ...         initialXYZ energy calculation done ...
 Status: 11 run mDynQ011 ...         start energyOptimization ...
 Status: 12 run mDynQ011 ...         mdSnap :  1 is wrote ...
 Status: 13 run mDynQ011 ...         energyOpimization is done ...
 Status: 16 run mDynQ011 ...         start molDyn run ...
 Status: 17 run mDynQ011 ...         mdSnap :  1 is wrote ...
 Status: 18 run mDynQ011 ...         mdSnap :  2 is wrote ...
 Status: 19 run mDynQ011 ...         mdSnap :  3 is wrote ...
 Status: 20 run mDynQ011 ...         mdSnap :  4 is wrote ...
 Status: 21 run mDynQ011 ...         mdSnap :  5 is wrote ...
 Status: 22 run mDynQ011 ...         mdSnap :  6 is wrote ...
 Status: 23 run mDynQ011 ...         mdSnap :  7 is wrote ...
 Status: 24 run mDynQ011 ...         mdSnap :  8 is wrote ...
 Status: 25 run mDynQ011 ...         mdSnap :  9 is wrote ...
 Status: 26 run mDynQ011 ...         mdSnap :  10 is wrote ...
 Status: 27 run mDynQ011 ...         mdSnap :  10 is wrote ...
 Status: 28 run mDynQ011 ...        mdRunCall Finish: StepDone=ntimeS: 2004
 Status: 29 run mDynQ011 ...         eqvilibration mDyn is done ...
 Status: 30 run mDynQ011 ...         successful finish mDynSB program ...
#
```

****

TEST#3 in ./t3 directory

Job3:
1) read snapshot pdb file from TEST2 calculation
2) make energy optimization

3) restrain positions of ProteinBackBone atoms with harmonic force field
4) make MD equilibration
5) make MD simulated annealing by protocol in file t3_SAprotocol.inp

test3 is running by command

```
>  $MDYN011HOME/mDynQ011 -i t3_MdynPar.inp -c 1arb.t3.inPdb.pdb
    -r1 t3_restrAt1.inp  -sa t3_SAprotocol.inp -mn 1arb -o t3.out

#files to run test3:
1) t3_MdynPar.inp:
#234567890123456789012345678901234567890!comment
$fullProtMD
$harmAt1PosRst=0.10                     !harmConst=0.1 (kcal/A^2)
$Hread
$shake=2                                !0/1/2! 2=all bonds are kept fixed
$SolvGS
$engCalc
$engOptim
$nOptStep=1                             !max N optim steps
$doMDyn
$MDSA
$initMDTemp=10.00
$bathMDTemp=50.00
$runMDnstep=500
$mdTimeStep=0.002
$nwtra=250
#END
----------------
2) file t3_SAprotocol.inp :
#SA protocol
#each line start from keyword SAPROT
#      ntimeMX    tempTarget SCvdW wfHb128BB wfhB128BS
SAPROT 1000        100.0     1.0    1.0     1.0
SAPROT 1000        300.0     1.0    1.0     1.0
SAPROT 1000        100.0     1.0    1.0     1.0
SAPROT 1000        50.0      1.0    1.0     1.0
END
-----------------------------------------------------------------------------
-------
3) file t3_restrAt1.inp :
#harmonically restrained RESidue segments
#(6x,2i4,a40)
RESTAT   1 263   PBB               !PBB - ProtBackBone atoms are restrained,
i.e. sideChain atoms are not
end
restrained
                                 !ALL - all atoms of residues are restrained
-----------------------------------------------------------------------------
-------
consol run out
Status: 1 run mDynQ011 ...
 Status: 2 run mDynQ011 ...         made molec topology ...
 Status: 3 run mDynQ011 ...         made [solvated] molec topology ..
 Status: 4 run mDynQ011 ...         init ForceField parameters ..
 Status: 5 run mDynQ011 ...         init Gauss Shell solvation model ..
 Status: 6 run mDynQ011 ...         mdSnap :  1 is wrote ...
 Status: 7 run mDynQ011 ...         initialXYZ energy calculation done ...
 Status: 8 run mDynQ011 ...         start energyOptimization ...
 Status: 9 run mDynQ011 ...         mdSnap :  1 is wrote ...
 Status: 10 run mDynQ011 ...         energyOpimization is done ...
 Status: 13 run mDynQ011 ...         start molDyn run ...
 Status: 14 run mDynQ011 ...         mdSnap :  1 is wrote ...
 Status: 15 run mDynQ011 ...         mdSnap :  2 is wrote ...
 Status: 16 run mDynQ011 ...         mdSnap :  3 is wrote ...
 Status: 17 run mDynQ011 ...         mdSnap :  4 is wrote ...
 Status: 18 run mDynQ011 ...         mdSnap :  5 is wrote ...
 Status: 19 run mDynQ011 ...         mdSnap :  6 is wrote ...
 Status: 20 run mDynQ011 ...         mdSnap :  7 is wrote ...
 Status: 21 run mDynQ011 ...         mdSnap :  8 is wrote ...
 Status: 22 run mDynQ011 ...         mdSnap :  9 is wrote ...
 Status: 23 run mDynQ011 ...         mdSnap :  10 is wrote ...
 Status: 24 run mDynQ011 ...         mdSnap :  10 is wrote ...
 Status: 25 run mDynQ011 ...        mdRunCall Finish: StepDone=ntimeS: 2004
 Status: 26 run mDynQ011 ...         eqvilibration mDyn is done ...
 Status: 27 run mDynQ011 ...         mdSnap :  11 is wrote ...
 Status: 28 run mDynQ011 ...         mdSnap :  12 is wrote ...
 Status: 29 run mDynQ011 ...         mdSnap :  13 is wrote ...
 etc.
 Status: 30 run mDynQ011 ...         successful finish mDynSB program ...
*****************************************************************************
```

TEST#4 in ./t4 directory

Job4:
1) read PDB file with H atoms
2) add positional restraints (harmonic force field) for defined atoms: file t4_restrAt1.inp
3) make energy optimization and molDyn
   for list of residues shown in file t4_moveRes.inp
4) run simulated annealing via protocol in file: t4_SAprotocol.inp

run TEST4 by command

```
> $MDYN011HOME/mDynQ011 -c 1arb.t4.InPdb.pdb -i t4_MdynPar.inp
   -sa t4_SAprotocol.inp -mv t4_moveRes.inp
   -r1 t4_restrAt1.inp -r2 t4_restrAt2.inp -rB t4_rigBody.inp -mn 1arb
    -o t4.out
```

```
#t4_MdynPar.inp
#234567890123456789012345678901234567890!comment
$MovingRes
$Hread
$hBond128 = 1.5
$compactForce = 0.25
$rigBody
$harmAt1PosRst=0.05                       !harmConst (kcal/A^2)
$distRestrA2
$shake=1                                  !0/1/2
$SolvMod = GShell
$engCalc
$engOptim
$nOptStep=1
$doMDyn
$MDSA                                     !do SimAnnealing
$initMDTemp=10.00
$bathMDTemp=50.00
$runMDnstep=500
$mdTimeStep=0.002
$NTV=1
$nwtra=250
#END
-------------
#t4_moveRes.inp : 1arb
aaaaaaIIIIiiii
#
MOVRES  91 179
MOVRES 190 240
end
------------
```

```
#t4_restrAt1.inp
#harmonically restrained RESidue segments
#xxxxxIIIIiiiiaaAAA
#(6x,2i4,a40)
RESTAT   1  63  ALL
RESTAT  64 179  PBB
RESTAT 200 250  PBB
end
-----------
# t4_restrAt2.inp
#harmonically restrained Atom-Atom distances
#1arb
#xxxxxx
#keyword atom1       atom2       distA HarmConst(kcal/mol*A^2)
RESTA2   ND2  ASN 222 : OG1 THR 219 = 7.0   1.5
RESTA2   O  GLY 170 : OG1 THR 219 = 8.0   2.5
RESTA2   OH TYR 109 : OG1 THR 111 = 7.5   3.0
END
------------------------------------------------
#t4_rigBody.inp : 1arb
aaaaaaIIIIiiii
#
RIGB01  91 179
RIGB02 190 240
END
-------------------------------------------------
#t4_SAprotocol.inp
#SA protocol
#nSAstep
#4
#(f10.1,1x,f8.1,1x,3(f6.1,1x)
#234567890x12345678x123456x123456x123456
#ntimeMX           tempTg  SCvdW wfHb128BB wfhB128BS
SAPROT 1000        100.0     1.0    1.0     1.0
SAPROT 1000        300.0     1.0    1.0     1.0
SAPROT 1000        100.0     1.0    1.0     1.0
SAPROT 1000        50.0      1.0    1.0     1.0
END
-----------
test4 out to console:

 Status: 1 run mDynQ011 ...
 Status: 2 run mDynQ011 ...         made molec topology ...
 Status: 3 run mDynQ011 ...         made [solvated] molec topology ..
 Status: 4 run mDynQ011 ...         init ForceField parameters ..
 Status: 5 run mDynQ011 ...         init Gauss Shell solvation model ..
 Status: 6 run mDynQ011 ...         mdSnap :  0 is wrote ...
 Status: 7 run mDynQ011 ...         initialXYZ energy calculation done ...
 Status: 8 run mDynQ011 ...         start energyOptimization ...
 Status: 9 run mDynQ011 ...         engOptimization step =  1 is done ...
 Status: 10 run mDynQ011 ...         mdSnap :  0 is wrote ...
 Status: 11 run mDynQ011 ...         energyOpimization is done ...
 Status: 14 run mDynQ011 ...         start molDyn run ...
 Status: 15 run mDynQ011 ...         mdSnap :  1 is wrote ...
 Status: 16 run mDynQ011 ...         mdSnap :  2 is wrote ...
 Status: 17 run mDynQ011 ...         mdSnap :  2 is wrote ...
 Status: 18 run mDynQ011 ...        mdRunCall Finish: StepDone=ntimeS: 504
 Status: 19 run mDynQ011 ...         eqvilibration mDyn is done ...
 Status: 20 run mDynQ011 ...         mdSnap :  3 is wrote ...
 Status: 21 run mDynQ011 ...         mdSnap :  4 is wrote ...
 Status: 22 run mDynQ011 ...         mdSnap :  5 is wrote ...
 Status: 23 run mDynQ011 ...         mdSnap :  6 is wrote ...
 Status: 24 run mDynQ011 ...         mdSnap :  6 is wrote ...
 Status: 25 run mDynQ011 ...        mdRunCall Finish: StepDone=ntimeS: 1004
 Status: 26 run mDynQ011 ...         mdSnap :  7 is wrote ...
 Status: 27 run mDynQ011 ...         mdSnap :  8 is wrote ...
 Status: 28 run mDynQ011 ...         mdSnap :  9 is wrote ...
 Status: 29 run mDynQ011 ...         mdSnap :  10 is wrote ...
 Status: 30 run mDynQ011 ...         mdSnap :  10 is wrote ...
 Status: 31 run mDynQ011 ...        mdRunCall Finish: StepDone=ntimeS: 1004
 Status: 32 run mDynQ011 ...         mdSnap :  11 is wrote ...
 Status: 33 run mDynQ011 ...         mdSnap :  12 is wrote ...
 Status: 34 run mDynQ011 ...         mdSnap :  13 is wrote ...
 Status: 35 run mDynQ011 ...         mdSnap :  14 is wrote ...
 Status: 36 run mDynQ011 ...         mdSnap :  14 is wrote ...
 Status: 37 run mDynQ011 ...        mdRunCall Finish: StepDone=ntimeS: 1004
 Status: 38 run mDynQ011 ...         mdSnap :  15 is wrote ...
 Status: 39 run mDynQ011 ...         mdSnap :  16 is wrote ...
 Status: 40 run mDynQ011 ...         mdSnap :  17 is wrote ...
 Status: 41 run mDynQ011 ...         mdSnap :  18 is wrote ...
 Status: 42 run mDynQ011 ...         mdSnap :  18 is wrote ...
 Status: 43 run mDynQ011 ...        mdRunCall Finish: StepDone=ntimeS: 1004
 Status: 44 run mDynQ011 ...         simulated annealing is done ...
 Status: 45 run mDynQ011 ...         successful finish mDynSB program ...

***********************
```

4. Performance

CPU time = 9-10 min/1000 MD step [athlon 1400 MHz]

for protein ~ 3000 atoms

II. Program flow and Basic algorithms of the program

1. Main program

```
 Main Program file  :       MDynSBmain.f
```

Start from the call of the input parameters

1.   call inputMDSApar

reads the main Input file
filenam = './MdynPar.inp'   ! in current job_dir

the file has the fixed name and located in the current job directory
the main input file **MdynPar.inp** defines main parameters of the job (see chapter input file description)

2.   call initMolecTopSeq01

reads a defined molecular PDB file, which can be defined in the **MdynPar.inp** file
or has the standard name ./molec.pdb and located in the current job directory ./ ;
defines residue sequence

3. call initMolecTopSeq02

calculates 12neighbour list (covalent bonds connecting atoms) using a predefined topology
information about resdues stored in the $MDSBHOME/dat

the pair12 list array: pair12List(*) is the basic molecular topology information.
Based on the pair12List(*) the all other lists are calculated, namely
Bonded triplets and quartets to form list of covalent angles, torsion angles, improper torsion angles.
The list of triplets and quartets are calculated via tree algorithm

```
       Call       vbondListPDB2(atomXYZ,
      &           natom,atomNumb,atomName,resName,chName,resNumb,
      &           nres,resNameRes,chNameRes,
      &           atomNameEx,startAtInRes,
      &           nmoveatom,moveAtomList,
      &           pair12List,startPairL12,nPairL12,np12MAX,
      &           pair13List,startPairL13,nPairL13,np13MAX,
      &           pair14List,startPairL14,nPairL14,np14MAX,
      &           bond12List,nbond12,
      &           trip123List,nTrip123,np123MAX,
      &           quar1234List,nQuar1234,np1234MAX,
      &           quarImp1234L,nImp1234,nImp1234MAX)
```

the call of the subroutine initMolecTopPDB results in the complete definition of the molecular topology
from the input molec.pdb 3D structure.

4. call initFFieldParam

Initialization of the force field parameters for the bond, angle, torsion angle, improper angle deformations,
van der waals non bond interactions and atomic point charges for the electrostatic interactions.
For bond, angle, torsion and improper angles a respective list of parameters are generated and stored in the arrays.

A list All force field parameters are based on the amber94 force field parameter set [Cornell et.al 1995].
Molecular mechanical energy is based on the standard equations for the force field of second generation
amber94 [Cornell et.al 1995].
Decoding of the atom names (residue names) to the forceField atom name is based on the look up table
ffAtomTypeFile = $MDSBHOME/dat/**atmAAmberff.dat**

5. Extraction of the data from Library file

All search of the proper names in the look up table of the MDynSB program are based on
the **hashing** of a records in the look up table, i.e. conversion of the table in numerically
sequential order. If several records of the look up table have the same hash number (degenerated case),
they are placed in a linkedLis for this hash number.

Force field parameters are taken from the file:
ffParFile = $MDSBHOME/dat/**bsparBATV.dat**
code fragment to initialize force field parameters
c get ff-atom code from atomNames

```
         call defFFatomName (ffAtomTypeFile,
     &              natom,atomNameEx,ResName,chName,
     &              ffAtomName,atomQ)
c
c define bondDef parameters for pair12List()
c
        call getBondDefPar(ffParFile,
     &              natom,atomNameEx,ResName,chName,ffAtomName,
     &              bond12List,nbond12,bond12ParL)
c c define valence angles def parameters
        call getVangDefPar(ffParFile,
     &              natom,atomNameEx,ResName,chName,ffAtomName,
     &              trip123List,nTrip123,ang123ParL)

c define Improper angle def parameters
        call getImpDefPar(ffParFile,
     &              natom,atomNameEx,ResName,chName,ffAtomName,
     &              quarImp1234L,nImp1234,impAng1234ParL)



c define torsion parameters
        call getTorsPar(ffParFile,
     &        natom,atomNameEx,ResName,chName,ffAtomName,
     &        quar1234List,nQuar1234,quar1234ParL,quar1234nPar)
c
c assign atomMass and vdwParameters
        call  getVDWatMass(ffParFile,
     &              natom,atomNameEx,ResName,chName,ffAtomName,
     &              nVDWtype,atomVDWtype,atomVDW12ab,atomMass)
c
c all FField Parameters are defined
```

6.   call initSolvatGSmod

Defines atomic parameters of the current structure for the Gaussian Shell implicit solvation model [Lazaridis, 1999].
A parameters of the GS model are stored in the files:

```
       solvGSPar_aa_amb.dat
       solvGSPar.dat
```

7.   call initMDStart(tempT0)

Initialize MD calculation:

Calculate the Initial nonBondPair lists
c generate three nonbonded atom pair Lists: van der Waals, Coulombic and solvation model.

```
c
        makeVdW = 1
        makeCL = 1
        makeSL = 1
c
        call initNonBondList(atomXYZ,makeVdW,makeCL,makeSL)
c
```

Calculates the forces on atoms for initial atomic coordinates
initial forces on atoms

```
c
        fcall = 0
       call initAllForce(fcall,atomXYZ,makeVdW,makeCL,makeSL,
     &              eVbondDef,vbdefForce,
     &              eVangDef,vAngdefForce,
     &              eImpDef,impDefForce,
     &              eTorsDef,torsAngForce,
     &              engVDWR1,vdwForceR1,
     &              engCOULR1,coulForceR1,
     &              engCOULR2,coulForceR2,
     &              restr1Eng,restr1AtForce,
     &              molSolEn, atomSolEn,atomSolFr)
c
```

Calculates initial atomic velocities, which are distributed according to Maxwell law

probability(v_i) = ( ) exp($-m_i v_i^2/kT$)

```
c
        call initVelocity(temp,natom,
     &       nmoveatom,moveAtomList,atomMass,atomVel0)
c
```

8. Run MD

The subroutine mdRun perform MD run for a given number of time steps ntimeMX

```
c
        call mdRun(ntimeMX,ntime0,ntime,ntimeR1,ntimeR2,
     &             ntimeF1,ntimeF2,ntimeF3,deltat,
     &             tempTg,tauTRF,atype,optra,wtra,nwtra,cltra)
c
```

9. Simulated Annealing optimization

```
c
        call simAnnealing(nSAstep,SAProtcol)
c
```

with user defined SAProtocol(nstep,T) consisted of nSAstep.

Each step of the SA is MD run of nstep with particular temperature T.

III. Details of the atomic force calculation

All atoms of the molecular system consists of two sets of **fixed** and **moving** atoms.

The force are calculated only for the moving atom set.

1. Covalent bond deformation

For covalent bond deformation we use the GROMOS functional form

$$\begin{aligned} V^{bond}(\mathbf{r}_1,\ldots,\mathbf{r}_N) &= \sum_{n=1}^{N_b} \frac{1}{4} K_{bn}[b_n^2 - b_{0n}^2]^2 \\ &= \sum_{n=1}^{N_b} V_n^{bond} \end{aligned} \tag{1}$$

where

rij = ri - rj

bn = rij .

This functional form is equivalent to the usual harmonic function for a small deformations but a computationally is more effective.

Force on atom i due to bond **n**

$$\begin{aligned} \mathbf{f}_{in} &= -\frac{\partial V_n^{bond}}{\partial b_n^2}\frac{\partial b_n^2}{\partial \mathbf{r}_i} = -K_{bn}[b_n^2 - b_{0n}^2]\mathbf{r}_{ij} \\ \mathbf{f}_{jn} &= -\mathbf{f}_{in} \end{aligned} \tag{2}$$

Total bond deformation force on atom i is the sum over all bonds **n** involving the atom i.

The calculation of the force $\mathbf{f}_{in}$ is doing by

```
subroutine vbonddefenf(xyz1,xyz2,bondPar,edef,f1,f2) (see file vdefenforce.f)
```

2. Covalent angle deformation

The covalent angle deformation energy function has the form

$$\begin{aligned} V^{angle}(r_1,\ldots,r_N) &= \sum_{n=1}^{N_{angle}} V_n^{angle}(\theta_n, K_{\theta_n}, \theta_{n_0}) \\ V_n^{angle}(\theta_n, K_{\theta_n}, \theta_{n_0}) &= \frac{1}{2} K_{\theta_n}[\cos\theta_n - \cos\theta_{n_0}]^2 \end{aligned} \tag{3}$$

This functional form is equivalent to the usual harmonic function for the angles for a small angle deformation but a computationally is more effective. The angle 2n (at the j) is between atoms i-j-k . The cosine of the angle 2n

$$\cos\theta_n = \frac{\mathbf{r}_{ij} \bullet \mathbf{r}_{kj}}{\left|\mathbf{r}_{ij}\right|\left|\mathbf{r}_{kj}\right|} \tag{4}$$

The forces on atoms i,j,k due to the deformation of the angle 2n

$$\begin{aligned}\mathbf{f}_i &= -\frac{\partial V_n^{angl}}{\partial \cos\theta_n}\frac{\partial \cos\theta_n}{\partial \mathbf{r}_i} \\ &= -K_{\theta_n}[\cos\theta_n - \cos\theta_{0n}][\frac{\mathbf{r}_{kj}}{r_{kj}} - \frac{\mathbf{r}_{ij}}{r_{ij}}\cos\theta_n]\frac{1}{r_{ij}}\end{aligned} \tag{5}$$

respectively force on atom k

$$\begin{aligned}\mathbf{f}_k &= -\frac{\partial V_n^{angl}}{\partial \cos\theta_n}\frac{\partial \cos\theta_n}{\partial \mathbf{r}_k} \\ &= -K_{\theta_n}[\cos\theta_n - \cos\theta_{0n}][\frac{\mathbf{r}_{ij}}{r_{ij}} - \frac{\mathbf{r}_{kj}}{r_{kj}}\cos\theta_n]\frac{1}{r_{kj}}\end{aligned} \tag{6}$$

force on atom j is given from the conservation of the total force acting on three atoms

$$\mathbf{f}_j = -\mathbf{f}_i - \mathbf{f}_k \tag{7}$$

The covalent angle deformation energy and force are calculated in subroutine

```
subroutine vangldefenf(xyz1,xyz2,xyz3,angPar,
     &                             edef,f1,f2,f3)
```

```
(see file vdefenforce.f)
```

3. Torsion angle energy and force

The total torsion energy is a sum over a set of torsion angles for the four atoms i-j-k-l with a rotation around bond j-k ,

$$V^{tors}(\mathbf{r}_1,\ldots,\mathbf{r}_N) = \sum_{n=1}^{N_t} V_n^{tors}(\varphi_n; torsPar)$$

$$V_n^{tors}(\varphi_n; torPar) = \sum_{\alpha=1}^{n_\alpha} K_{n\alpha}[1+\delta_\alpha \cos(m_\alpha \varphi_n)] \tag{8}$$

where torsion energy for bond j-k can have several torsion barriers with different multiplicity.

Torsion angle N is defined as

$$\phi = sign(-\mathbf{r}_{jk} \cdot (\mathbf{r}_{ij} \times \mathbf{r}_{kl})) \cdot \arccos(\frac{\mathbf{r}_{im} \cdot \mathbf{r}_{\ln}}{r_{im} r_{\ln}})$$

$$\cos\phi = \frac{\mathbf{r}_{im} \cdot \mathbf{r}_{\ln}}{r_{im} r_{\ln}} \tag{9}$$

where

$$\mathbf{r}_{im} = \mathbf{r}_{ij} - \frac{(\mathbf{r}_{ij} \bullet \mathbf{r}_{kj})}{r_{kj}^{\;2}} \mathbf{r}_{kj} \tag{10}$$

$$\mathbf{r}_{\ln} = -\mathbf{r}_{kl} + \frac{(\mathbf{r}_{kl} \bullet \mathbf{r}_{kj})}{r_{kj}^{\;2}} \mathbf{r}_{kj} \tag{11}$$

The forces on atoms i,j,k,l due to the single term of eq.(8b) are

$$\begin{aligned}\mathbf{f}_i &= -\frac{\partial V_{n\alpha}^{tors}}{\partial \mathbf{r}_i} = -\frac{\partial V_{n\alpha}^{tors}}{\partial \cos(m_\alpha \varphi_n)} \frac{\partial \cos(m_\alpha \varphi_n)}{\partial \cos(\varphi_n)} \frac{\partial \cos(\varphi_n)}{\partial \mathbf{r}_i} \\ &= -K_{n\alpha}\delta_\alpha \frac{\partial \cos(m_\alpha \varphi_n)}{\partial \cos(\varphi_n)} [\frac{\mathbf{r}_{\ln}}{r_{\ln}} - \frac{\mathbf{r}_{im}}{r_{im}} \cos\varphi_n] \frac{1}{r_{im}}\end{aligned} \qquad (12)$$

$$\begin{aligned}\mathbf{f}_l &= -\frac{\partial V_{n\alpha}^{tors}}{\partial \mathbf{r}_l} = -\frac{\partial V_{n\alpha}^{tors}}{\partial \cos(m_\alpha \varphi_n)} \frac{\partial \cos(m_\alpha \varphi_n)}{\partial \cos(\varphi_n)} \frac{\partial \cos(\varphi_n)}{\partial \mathbf{r}_l} \\ &= -K_{n\alpha}\delta_\alpha \frac{\partial \cos(m_\alpha \varphi_n)}{\partial \cos(\varphi_n)} [\frac{\mathbf{r}_{im}}{r_{im}} - \frac{\mathbf{r}_{\ln}}{r_{\ln}} \cos\varphi_n] \frac{1}{r_{\ln}}\end{aligned} \qquad (13)$$

$$\mathbf{f}_j = [\frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{kj}}{r_{kj}^2} - 1]\mathbf{f}_i - \frac{\mathbf{r}_{kl} \cdot \mathbf{r}_{kj}}{r_{kj}^2}\mathbf{f}_l \qquad (14)$$

and finally

$$\mathbf{f}_k = -(\mathbf{f}_i + \mathbf{f}_j + \mathbf{f}_l) \qquad (15)$$

The torsion energy and force are calculated via

```
        subroutine torsanglenf(xyz1,xyz2,xyz3,xyz4,nTorsH,
     &                            torsPar,eTors,f1,f2,f3,f4)


c torsPar(4*nTorsH) = {pass,Vt/2/pass,cos(delta),nFi },…
c  eTors = sum{ Ki*[1+cos(delti)cos(i*Ftors)] }; i=1,..,nTorsH
c
```

Torsion parameters are taken from the LibData = bsparBATV.dat

The extraction of the torsion parameters from LibData = bsparBATV.dat for all quartets is done by

```
       subroutine getTorsPar(ffParFile,
     &              natom,atomNameEx,ResName,chName,ffAtomName,
     &              quar1234L,nQuar1234,quar1234Par,quar1234nPar)
c
c InPut:
c       ffParFile - ffParameters file
c       natom,atomNameEx,ResName,chName : PDB info
c       ffAtomName(ia) - FFatomName to search table
c       the quar1234L(i),i=1,..,nQuar1234 : the QuartetList
c RESULT: quar1234Par(16*nQuar1234) - torsionFF parameters for list
c         of  quartets
c         pass,Vt/2,delta,nFi - (printed) for each torsHarmonics,
c         pass,Vt/2/pass,cos(delta),nFi - finally in array
```

```
c        4- torsionHarmanics is possible.
c        quar1234nPar(iQuart) - number of torsHarmonics for the torsAngl
c
```

4. Improper Torsion Angle (out of plane) deformation

The improper torsion angle deformation keeps the four atoms 1-2-3-4 (i-j-k-l) in specified geometry. The first atom in the improper quartet is a planar or (tetrahedral) atom. For example atoms Ci-CAi-N(i+1)-Oi are kept planar. The out of plane potential

$$V^{imp}(\mathbf{r}_{1,\ldots,}\mathbf{r}_n) = \sum_{n=1}^{N_{imp}} V_n^{imp}(\xi_n;\xi_0,K_{\xi 0})$$

$$V_n^{imp}(\xi_n;\xi_0,K_{\xi 0}) = \frac{1}{2}K_{\xi 0}(\xi_n-\xi_0)^2 \tag{16}$$

CA-N-C-CB are kept in the tetrahedral configuration (L-amino acid) or CA-C-N-CB (D-amino acid) if CA in the united atom (CH) presentation.

The out of plane angle is defined for j-i-k four atoms with i is the planar (tetrahedral)

L

angle between to planes (i-j-k) and (j-k-l) with rotation angle around j-k, other words the

torsion angle in the sequence i-j-k-l

$$\xi_n = sign(\mathbf{r}_{ij}\cdot\mathbf{r}_{nk})\arccos(\frac{\mathbf{r}_{mj}\cdot\mathbf{r}_{nk}}{r_{mj}r_{nk}}) \tag{17}$$

where

$$\mathbf{r}_{mj} = \mathbf{r}_{ij}\times\mathbf{r}_{kj} \tag{18}$$

$$\mathbf{r}_{nk} = \mathbf{r}_{kj}\times\mathbf{r}_{kl} \tag{19}$$

The forces on atoms i,j,kl due to a single term Vn

$$\mathbf{f}_i = -\frac{\partial V_n^{imp}}{\partial \xi_n}\frac{\partial \xi_n}{\partial \mathbf{r}_i} = \\ -K_{\xi n}[\xi_n - \xi_0]\frac{r_{kj}}{r_{mj}^2}\mathbf{r}_{mj} \tag{20}$$

$$\mathbf{f}_l = -\frac{\partial V_n^{imp}}{\partial \xi_n}\frac{\partial \xi_n}{\partial \mathbf{r}_l} = \\ K_{\xi n}[\xi_n - \xi_0]\frac{r_{kj}}{r_{nk}^2}\mathbf{r}_{nk} \tag{21}$$

$$\mathbf{f}_j = -\frac{\partial V_n^{imp}}{\partial \xi_n}\frac{\partial \xi_n}{\partial \mathbf{r}_j} \\ = [\frac{\mathbf{r}_{ij}\cdot\mathbf{r}_{kj}}{r_{kj}^2} - 1]\mathbf{f}_i - \frac{\mathbf{r}_{kl}\cdot\mathbf{r}_{kj}}{r_{kj}^2}\mathbf{f}_l \tag{22}$$

finally from the third Newton law

$$\mathbf{f}_k = -(\mathbf{f}_i + \mathbf{f}_j + \mathbf{f}_l) \tag{23}$$

The improper energy and forces for a given improper quartet of atoms are calculated by the subroutine

```
c improper torsion energy force
c
        subroutine imprtorsanglenf(xyz1,xyz2,xyz3,xyz4,impPar,
     &                          eImpt,f1,f2,f3,f4)

c
c ImptPar(2) = K1, ksi0
```

5. Covalent back-bond deformation calculation

All valence back-bond deformation are calculated in the file initAllForce.f

```
subroutine initAllForce(fcall,atomXYZ,
     &              makeVdWs,makeCLs,makeSLs,
     &              eVbondDef,vbdefForce,
     &              eVangDef,vAngdefForce,
     &              eImpDef,impDefForce,
     &              eTorsDef,torsAngForce,
     &              engVDWR1,vdwForceR1,
     &              engCOULR1,coulForceR1,
     &              engCOULR2,coulForceR2,
     &              restr1Eng,restr1AtForce,
     &              molSolEn, atomSolEn, atomSolFr)
c
        include 'xyzPDBsize.h'
        include 'xyzPDBinfo.h'
        include 'pair1234array.h'
        include 'nbondPairVCS.h'
        include 'vdw12Par.h'
        include 'restrainInfo.h'
        include 'loopInfo.h'
        include 'movingAtom.h'
        include 'solvGSarray.h'
        include 'optionPar.h'
c
 . . . . . . . . . . . . . . . . . . . . .

c
c all GeoDef forces are calculated at each step

        call allAtVBondEForce(atomXYZ,
     &           natom,bond12List,nbond12,bond12ParL,
     &           eVbondDef,vbdefForce )
c
c
        call allAtVangEForce(atomXYZ,
     &           natom,trip123List,nTrip123,ang123ParL,
     &           eVangDef,vAngdefForce )
c
c
        call allAtImpTEForce(atomXYZ,
     &           natom,quarImp1234L,nImp1234,impAng1234ParL,
     &           eImpDef,impDefForce )
c
c torsionEnForces
c
        call allAtTorsEForce(atomXYZ,
     &           natom,quar1234List,nQuar1234,
     &           quar1234ParL,quar1234nPar,
     &           eTorsDef,torsAngForce )
c

 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . .
```

The deformation forces are calculated at each time step in the MD run.

6. Non bonded pair list calculation

The non bonded pair interactions are calculated for the pair list. Pair list for the central atom i is a sequence of atom numbers for atom within the radius R from the central atom. Three separate pair lists are calculated. The Van der Waals pair list(i) includes atom j if

$$r_{ij} \le R1+)R \qquad (24)$$

where)R is the buffer size. The buffer size defines the rate of pair list updating frequency

$$N_{UPDATE} =)R/[)tVmax] \qquad (25)$$

where Vmax is the maximal velocity of an atoms and)t is the time step. The optimal (over CPU time) value of the buffer size can be found. A default value is)R=1 Å.

The pair list calculated with via the lattice algorithm:

- a) the atomic coordinates $\mathbf{r}_1,\ldots,\mathbf{r}_N$ are projected on the cubic lattice, the integer coordinates of the atoms $\mathbf{h}_1,\ldots,\mathbf{h}_N$ are obtained. The lattice size is quite small ~ 2 A, to include just one atom.
-

The linked list and all pairList (nnbPairLV, nnbPairLC, nnbPairLS) are calculated in the subroutine

```
c
        subroutine nonbondListVCS(rcutV,rcutC,rcutS,atomXYZ,atomQ,
     &           rbuffV,rbuffC,rbuffS,
     &           makeVdW,makeCL,makeS,
     &           natom,atomNumb,atomName,resName,chName,resNumb,
     &           nres,resNameRes,chNameRes,
     &           atomNameEx,startAtInRes,
     &           nmoveatom,moveAtomList,moveFlag,
     &           pair12List,startPairL12,nPairL12,
     &           pair13List,startPairL13,nPairL13,
     &           pair14List,startPairL14,nPairL14,
     &           nbpairListV,startnbPairLV,nnbPairLV,nnbpLVMAX,
     &           nbpairListC,startnbPairLC,nnbPairLC,nnbpLCMAX,
     &           nbpairListS,startnbPairLS,nnbPairLS,nnbpLSMAX)
c
```

fragment of code for the linked list calculation:

```
c distribute atoms over cells
c make linked list of atoms in cells
c headat(n) - head(incellN)
c linkList(ia) - linkedList
        ixm=1
        iym=1
        izm=1
        do ia = 1,natom
c calculate cell numb
        i3=3*ia-3
        xyzi(1)=atomXYZ(i3+1)-xMIN(1)
        xyzi(2)=atomXYZ(i3+2)-xMIN(2)
        xyzi(3)=atomXYZ(i3+3)-xMIN(3)
        ix = xyzi(1)/cellh+1
        iy = xyzi(2)/cellh+1
        iz = xyzi(3)/cellh+1
        if(ixm .lt. ix)ixm = ix
        if(iym .lt. iy)iym = iy
        if(izm .lt. iz)izm = iz
c cell number
        ncell = ix + (iy-1)*nsiz(1) + (iz-1)*nsiz(1)*nsiz(2)
        if(ncell .gt. ncell3MAX)then
        write(kanalp,*)'ERROR!:nonbondList: ncell3MAX is low !!'
        stop
        end if!

c make linked list
        linkList(ia) =  headat(ncell)
        headat(ncell) = ia
        end do !ia
c end of linked list calculation

The pair lists VDW and COULOMbic energy exclude 12, 13, 14 covalent bonded
pairs. The Solvent model pairList
include all 12,13, 14 pairs.
The pair list are calculated for the range respectively:
c
        rcutV2 = (rcutV + rbuffV)**2     ! range for List1 -
                                                     VDWaals -  nbPairListV
        rcutV2m = (rcutV - rbuffC)**2    ! range for List2 - Coulombic twin
                                                        range - nbPairListC

        rcutC2p = (rcutC + rbuffC)**2    ! range for List2
        rcutS2 = (rcutS + rbuffS)**2     ! range for SolvationGSList -
                                                                nbPairListS
c

see file nonbobdListVCS.f
```

7. Non bonded force calculation

Van der waals forces are calculated for the non-bonded pair list nbpairListV()for atoms j within rij < RCUTV the cutoff radius for van der waals interactions. The modified potential 6-12 are used

$$U_{vdw} = \sum_{j=1}^{Nj} V_{6-12}^{s}(r_{ij}) \tag{26}$$

where the modified potential is a smoothed 6-12 for a small distances r

$$\begin{aligned} V_{6-12}^{s}(r) &= \frac{A12}{r^{12}} - \frac{B6}{r^{6}} \quad \text{if } \mathrm{r_{ij}} > r_s \\ &= \frac{\partial \mathrm{V}_{6\text{-}12}(r_s)}{\partial r}[r_{ij} - r_s] + \mathrm{V}_{6\text{-}12}(r_s) \ \text{ if } \mathrm{r_{ij}} < r_s \end{aligned} \tag{27}$$

the pair list for atom i includes atoms j > i, to count each pair interaction once. The force $\mathbf{F}^{vdwi}$ on atom i due to interaction with atoms in the pair list

$$\mathbf{F}_{i}^{vdw} = \sum_{j=1}^{Nj} \mathbf{f}_{ij} = \sum_{j=1}^{Nj} \frac{\partial V_{6-12}^{s}(r_{ij})}{\partial r_{ij}} \tag{28}$$

The modified (smoothed) 6-12 potential prevents over-flow when atoms are too close and generates smooth driving forces to resolve clash problems between atoms in molecular dynamics simulations, see

```
c
        subroutine vdwenforceij(dij2,dij1,rij,A12,B12,evdw,fi)
c
```

The coulombic energy and forces for atom i are calculated for all pairs within the radius RCUTC.

The coulombic energy/forces for a central atom i are calculated for the classical coulombic law or as a coulombic interaction between two charges on the compensating background charge uniformly distributed within the sphere of radius RCUTC

$$v_{cl}(r_{ij}) = \frac{q_i q_j}{r_{ij}} \qquad (29)$$

The modified electrostatic potential on the compensating background charge

$$v_{ucl}(r_{ij}) = \frac{q_i q_j}{r_{ij}}\left(1 + \frac{r_{ij}^3}{2R_c^3} - \frac{3r_{ij}}{2R_c}\right)\Theta(R_c - r_{ij}) \qquad (30)$$

has zero interaction energy and forces for the rij > RCUTC. This form of electrostatic interactions is better suitable to prevent energy conservation in the molecular dynamic calculation, see

```
c
        subroutine coulenforceij(var,rcutC,dij2,dij1,rij,qi,qj,ecoul,fi)
c
```

The nonbonded energy and force within short range RCUTV=R1 are calculated in the subroutine

```
c allAtNonBondEForce : VDW and COULOMBIC
c
        subroutine allAtVDWEForceR1(atomXYZ,atomQ,
     &           natom,nmoveatom,moveAtomList,
     &           nbpairListV,startnbPairLV,nnbPairLV,
     &           pair14List,startPairL14,nPairL14,
     &           nVDWtype,atomVDWtype,atomVDW12ab,
     &           rcutV,rcutC,engVDW,vdwForce,engCOULR1,coulForceR1)
c
```

for the pair list nbpairListV() and pair14List(). The last one includes all 1-4 neihgbours for which the **amber** force field uses the scaling factors for van der waals and coulombic interactions.

To increase performance of the van der waals energy/force calculations the table of coefficient A12, B12 for all atom types are precalculated and then right values A12/B12 for a given atom types in the pair ij are extracted from the vdw AB-parameter table

```
c get pointer to the AB table
        call vdw12TablePos(nVDWtype,t1,t2,t12)
        p4 = 4*t12
        A12 = atomVDW12ab(p4-3)
        B12 = atomVDW12ab(p4-2)
c
```

The long-range electrostatic forces within RCUTV < rij < RCUTC are calculated via the subroutine

```
c
        subroutine allAtVDWEForceR2(atomXYZ,atomQ,
     &           natom,nmoveatom,moveAtomList,
     &           nbpairListC,startnbPairLC,nnbPairLC,
     &           rcutR1,rcutR2,engCOULR2,coulForceR2)
c
c LongRamge -   RCUT1 < rij < RCUT2
```

The program keep separately the short-range and the long-range electrostatic energy and force.

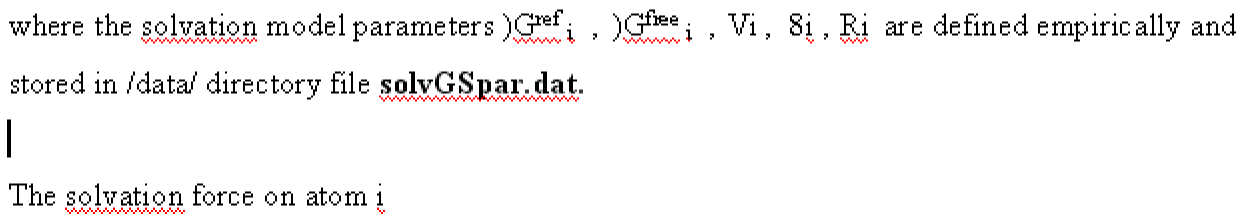
8. Solvation energy/force calculation

The implicit solvation model - the Gaussian Shell model of Lazaridis & Karplus is used to calculate the solvation energy [POTEINS 35: 133-152, 1999]. The solvation free energy of the atom i

$$\Delta G_i^{sl} = \Delta G_i^{ref} - \sum_{j \neq i} g_i(r_{ij}) V_j \qquad (31)$$

where sum is going over all neighbors of atom i which exclude volume Vj from the solvation volume around of the atom i. The function gi(r) describe the solvation energy density in the volume around the atom i and is approximated by the Gaussian function

$$g_i(r) = \frac{\Delta G_i^{free}}{2\pi\, r^2 \sqrt{\pi}\, \lambda_i} \exp(-[\frac{r - R_i}{\lambda_i}]^2) \qquad (32)$$



where the solvation model parameters)Gref i ,)Gfree i , Vi , 8i , Ri are defined empirically and stored in /data/ directory file **solvGSpar.dat.**

The solvation force on atom i

$$\mathbf{f}_i = -\frac{\partial G^{sl}}{\partial \mathbf{r}_i} = -\sum_{j \neq i} g_i(r_{ij})[\frac{r_{ij} - R_i}{\lambda_i^2} + \frac{1}{r_{ij}}]\frac{V_j}{r_{ij}}(\mathbf{r}_i - \mathbf{r}_j)$$
$$- \sum_{j \neq i} g_j(r_{ij})[\frac{r_{ij} - R_j}{\lambda_j^2} + \frac{1}{r_{ij}}]\frac{V_i}{r_{ij}}(\mathbf{r}_i - \mathbf{r}_j) \qquad (33)$$

The sum over all solvation forces **fi** is zero.

The solvation forces are calculated by subroutine

```
c
                 call SolventEnForces(natom, atomXYZ,
     &         atomName,startPairL12,nPairL12,pair12List,
     &         nbpairListsS,startnbPairLS,nnbPairLS,
     &         atomSolPar, molSolEn, atomSolEn, atomSolFr)
c
```

IV. Details of MD run

An MD run is performed by subroutine

```
c
        subroutine mdRun(ntimeMX,ntime0,ntime,ntimeR1,ntimeR2,
     &               ntimeF1,ntimeF2,ntimeF3,deltat,
     &               tempTg,tauTRF,atype,optra,wtra,nwtra,cltra)
c
c MD RUN propagates MDtraj from files in mdAtomXYZvel.h
c                                    [ atomXYZ0(*),atomVel0(*) ]
c    call initMDStart(T) inits the MD start
c                         from the INput atomXYZ(*)-->atom0XYZ(*)
c
c ntimeMX max number of time steps
c ntime0 - executed number of timesteps in the previous call
c ntime   executed number of timesteps in this call
c ntimeR1, ntimeR2 - update frequency  for R1, R2 pairLists
c ntimeF1,ntimeF2 - update freq for R1=(vdw+coulR1), R2-coulR2 en/forces
c ntimeF3 -  SOLVation forces
c GeoEn/force ntimeFg=1 - standart
c deltat- timestep, temp - initial(temp) of MD run
c tempTg - target T for NTV ansemble[K]
c tauTRF - tau Relaxation Factor [ps]
c atype - ansamble type = 0/1 - NEV, NTV
```

The MD algorithm consist of a long loop over the time steps.

For each time step MD trajectory is propagated for the)t = 1-2 femto sec, as defined by user.

1. Pair lists

The pair lists are updated for each n-th timestep equal to ntimeR1, ntimeR2 for the short-range and for the twin-range long-range electrostatic energy calculations.

```
c
        call initNonBondList(atomXYZ0,makeVdW,makeCL,makeSL)
c
```

2. The atomic forces

The atomic forces due to deformation of covalent structure and short-range non-bonded calculation are updated for the each ntimeF1-th time step, the long-range electrostatic are updated for the each ntimeF2-th step and solvation forces are updated for each ntimeF3-th time step.

{Note! In the current version the multiple time step for pair list update and md equation integration are equal. The general case is not tested !}

```
c update forces/energy
        call initAllForce(fcall,atomXYZ0,doVdWef,doCLef,doSLef,
     &              eVbondDef,vbdefForce,
     &              eVangDef,vAngdefForce,
     &              eImpDef,impDefForce,
     &              eTorsDef,torsAngForce,
     &              engVDWR1,vdwForceR1,
     &              engCOULR1,coulForceR1,
     &              engCOULR2,coulForceR2,
     &              restr1Eng,restr1AtForce,
     &              molSolEn, atomSolEn, atomSolFr)
```

MD simulation can be done with a specified set of forces. The set of forces can be specified by the array fEngWF(*)

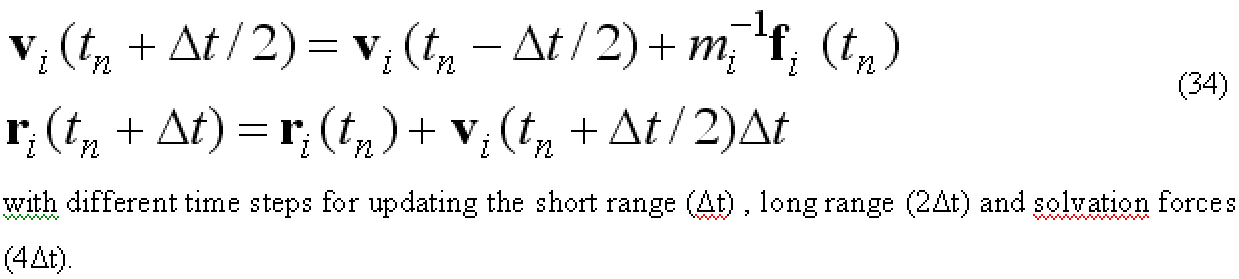
```
c
          eGeoDef   = fEngWF(1)*eVbondDef + fEngWF(2)*eVangDef
     &                + fEngWF(3)*eImpDef + fEngWF(4)*eTorsDef
     &                + fEngWF(8)* restr1Eng
          engCOUL   = fEngWF(6)*engCOULR1 + fEngWF(7)*engCOULR2
          engPOTENT = eGeoDef + fEngWF(5)*engVDWR1 + engCOUL +
     &                molSolEn*fEngWF(9)
c
```

3. Propogation of the trajectory

For one time step propagation of the MD trajectory is done by the subroutine

```
c make mdStep
        call mdTimeStepProp(nmoveatom,moveAtomList,deltat)
c
```

which uses multi step leap-frog algorithm to calculate velocities and positions at time (t+deltat).



$$\mathbf{v}_i(t_n+\Delta t/2)=\mathbf{v}_i(t_n-\Delta t/2)+m_i^{-1}\mathbf{f}_i\ (t_n)$$
$$\mathbf{r}_i(t_n+\Delta t)=\mathbf{r}_i(t_n)+\mathbf{v}_i(t_n+\Delta t/2)\Delta t \qquad (34)$$

with different time steps for updating the short range (Δt) , long range (2Δt) and solvation forces (4Δt).

4. Temperature control - Berendsen thermostat method

At each time step the temperature control routine performs calculation of the total kinetic energy of the moving atoms. The relaxation the average temperature of the atomic system to the specified value are give via the *weak-coupling method* or Berendsen method, which scale the velocity by the factor lambTR(t)

$$V'_i(t) = V_i(t)^* \ \mathrm{lambTR}(t) \qquad (35)$$

the velocity scaling describes energy exchange with bath thermostat with temperature relaxation time τ_T. The respective scaling factor is equal

$$\mathrm{lambTR(t) = sqrt(1 + (tempTg-tempT0(t))/ \tau_T)*(tempTg/tempT0 -1.0))} \qquad (36)$$

where tempT0 is the effective temperature at the time=t, and tempTg is the target temperature to relax. The effective temperature tempT0(t) is defined by the all atomic velocities

$$T0(t) = \frac{1}{k_B N_{\deg Freed}} \sum_{i=1}^{Nat} m_i V_i^2(t) \qquad (37)$$

where $N_{degFreed}$ is the number degrees of freedom, k_B is the Boltzman constant. For proteins in water solvent a reasonable value of the temperature relaxation time τ_T is equal to 0.4-0.5 ps. The value of τ_T should be sufficiently small to achieve required temperature, but sufficiently large to avoid disturbance of the properties of protein by strong coupling to the temperature bath.

5. Trajectory writing

Trajectory is written for each nwtra time steps. The trajectory can be written for atomic positions (and for atomic velocietis) in the user specified file.

6. References:

Tamar Schlick. Molecular Modeling and simulation. Springer-Verlag, New York, 2000.
Cornell W.D., Cieplak P., Bayly C.I., Gould I.R., Mertz K.M., Ferguson D., Spellmeyer D.C., Fox T., Caldwell J.W., Kollam P.A. A second generation force field for the simulation of proteins, nucleic acids and organic molecules. J.Am.Chem.Soc. 1995: 117, p.5179-5197
Lazaridis T., Karplus M. Proteins: Structu, Funct., and Gen. 1999: 35, p.133-152

Parameters

Input file

Input file. The inProtocol file defines protocol of mdyn calculations.
Default file name ./MdynPar.inp .
inProtocol file consist of sequense of lines. Line starts from keyWord [and its value].
Example of inProtocol file:

```
#MdynPar.inp for HomologyModel refinement
#234567890123456789012345678901234567890!comment
$fullProtMD                                              !
#$MovingRes
$harmAt1PosRst=0.25                                      !harmConst
(kcal/A^2)
$Hread
$shake=2                                                 !0/1/2
$zeroRot
#$SolvateExWat=4.5                                       !ExplicitWaterShell
4.5A
#$SolvGS
```

```
$SolvWbrg
$SolvGBorn                                               !SolvGBorn
#$mdRestart
$doMDyn
$MDSA                                                    !do SimAnnealing
$engCalc
$engOptim
$nOptStep=1                                              !max N optim steps
$aSoftCore=1.0                                           ! 1.0= standart VDW,
< 1.0 -0.0-softCore
$initMDTemp=10.00                                        ! initial tempera-
ture in K
$bathMDTemp=50.00                                        ! bath termostat
temperature
$runMDnstep=2000                                         ! number mdyn time
step to run
$mdTimeStep=0.002                                        ! md time step
$NTV=1                                                   ! statistical ensem-
ble type NTV/NEV = 1/0
$nwtra=500                                               ! write on HD pro-
tein structur in pdb format for each nwtra mdstep
#
END
#
NOTE that parameter file formatted, i.e. $ sign should be in the firs posi-
tion of the line  No SPACE to assign value after keyword.
#
Description:
parameter file consists of lines starting from the $ simbol and keyWord
keyWord can be two types: logical  and digital
$MovingRes                 ! logical  required special file to define moving
RESidues list
$harmAt1PosRst=0.25        ! digital  NO SPACE to assign value for keyword

keyWord  switch on a respective modul of program,
some keyWord switch on moduls which in turn needs some special User defined
file to work properly.

KEYWORD DESCRIPTION
#234567890123456789012345678901234567890!comment

$fullProtMD                             !defines FULL (i.e. ALL atoms) of the
USER molecule
                                         will be free to move in energy re-
laxation or molDyn
$MovingRes                              ! logical keyWord defines that ONLY
a defined set of RESidue are free to move
                                          this keyWord is coupled with file
-mv moveRes in the argument line of
                                          the program mdynSB0
                                         default name for moveRes file is
./moveRes.inp
#example of ./moveRes.inp
#1arb
#aaaaaaIIIIiiii
#
MOVRES   1  10       !line defines first and last resudues of moving segment
MOVRES  45  76
MOVRES 115 260
end
************
$harmAt1PosRst=0.25  ! digital keyWord define RESidue segments with 1 atom
position harmonic restrants.
                                     0.25 = harmonic restrain Constant K
                                     restrEnergy = 0.5*K(r - r0)**2,
                                     the reference position r0 = initialXYZ-
input.pdb - positions from
                                     the initial INPut PDB file which defines
INItial structure of molecule

                                     this keyWord is coupled with file -r in-
Restrain of the argument line of
                                     the program mdynSB05
                                     default name for inRestrain file is
./restrAt1.inp

EXample of inRestrain file:
#harmonically restrained RESidue segments
#xxxxxIIIIiiiiaaAAAA
#(6x,2i4,a40)
RESTAT   1   63  PBB           ! line starts from keyWord RESTAT
numbers=first/last residue of segment
                               ! PBB (only protein backbone atoms are re-
strained, i.e. side chains are free)
RESTAT  78 120  ALL            ! ALL (all atoms are restrained)
end
#
 ---------------------------------------------------

$Hread    ! defines that all Hydrogens will be read from input molecule
structure -c inPDB   file
                otherwise the ALL HYDrogens will be restored by the program
mdynSB05
                RECOMENDED: at the first run of a protein with unknown (or
partially known) Hydrogen atom.
                               start the mdynSB with off $Hread option, i.e.
                               #$Hread
----------------------------------------------------
$shake=2    ! invoke shake subroutine to keep bonds fixed. shake=1  X--Hydr
bonds, (shake=2 all bonds) are fixed

-----------------------------------------------------------------------------
---------------------------
$zeroRot   ! invoke procedure to stop overal rotation and translation of mol-
ecule

----------------------------------------------------
$SolvateExWat=4.5   ! build explicit water solvation shell of 4.5 A around
protein molecule

----------------------------------------------------

$SolvGS         ! invoke implicit  Gaussian Shell solvation model
$SolvWbrg       ! implicit  WaterBridges between polar atoms
$SolvGBorn      ! implicit Generalized Born model + SAS HydroPhobic solvation

---------------------------------------------------

$mdRestart    ! restart molDynamics from the last snapshot  mdXYZVfin.pdb
                the file mdXYZVfin.pdb should be copied to the file mdyn in-
Restart file
                mdXYZVin.pdb

$doMDyn       ! do molecular dynamics
$MDSA         ! do Molecular Dynamical Simulated Annealing
              ! coupled with file -sa SAprotocol which define protocol of the
simulated annealing
```

```
Example of SAprotocol.inp  file
#SA protocol
#nSAstep
2
#(f10.1,1x,f8.1,1x,3(f6.1,1x)
#234567890x12345678x123456x123456x123456
#ntimeMX    tempTg  SCvdW wfHb128BB wfhB128BS
100000      500.0     0.8    1.0     1.0
100000      100.0     1.0    1.0     1.0
END
#
   ntimeMX - number of md timeStep
   tempTg    - target temperature in K, this temperature will be reach during
ntimeMX steps
   SCvdW     - parameter 0 - 1 to defile softness of the van der waals poten-
tial. Soft potential
             modifies Potential Energy Surface decrease a barriers of confor-
mational transitions
   wfHb128BB, wfhB128BS - scaling factors for BackBone-BackBone and BackBone-
SideChain Hydrogen Bond energy
#-----------------------------------------------------
```

```
---------------------------------------------------
* * * * * * * * * * * * * * * * * * * * * * * *
-c inPDB   file  - standart pdb file
REMARK: PDB:
ATOM      1  N    GLY A   1      11.726 -10.369  10.598
ATOM      2  H1   GLY A   1      11.921 -11.015   9.807
ATOM      3  H2   GLY A   1      12.518 -10.395  11.271
ATOM      4  H3   GLY A   1      10.852 -10.663  11.079
ATOM      5  CA   GLY A   1      11.567  -9.015  10.090
ATOM      6  HA2  GLY A   1      10.772  -8.977   9.420
ATOM      7  HA3  GLY A   1      12.439  -8.710   9.612
ATOM      8  C    GLY A   1      11.280  -8.099  11.303
ATOM      9  O    GLY A   1      11.256  -8.584  12.493
ATOM     10  N    VAL A   2      11.060  -6.876  11.020
ATOM     11  H    VAL A   2      11.066  -6.574  10.025
etc.
TER                 ! CHAIN TERmination
ATOM   1302  N   GLY A  94      10.957 -15.678  12.832
ATOM   1303  H   GLY A  94      10.735 -14.663  12.877
ATOM   1304  CA  GLY A  94      10.193 -16.559  11.950
ATOM   1305  HA2 GLY A  94       9.428 -16.004  11.516
ATOM   1306  HA3 GLY A  94       9.784 -17.323  12.525
ATOM   1307  C   GLY A  94      11.016 -17.184  10.843
...
etc.
TER                 ! CHAIN TERmination
END                 ! file  END
* * * * * * * * * * * * * * * * * * * * * * *
```

PDB file - inPDB file Default name ./molec.pdb

XYZ+Velocity data - The XYZ+Velocity file to REStart MolDyn simulation from the last snapshot.

Position Restrain data - File with definition of harmonic positional restraints for user-specified atoms in a molecule.

Distance Restrain data - File with definition of harmonic distance restraints for user-specified pairs of atoms in a molecule.

Rigidbody data - File with definition of harmonic distance restraints for user-specified pairs of atoms in a molecule.

saProtocol file:

```
saProtocol file . User defined protocol for simulated annealing molecular dynamics.
Default file name ./Saprotocol.inp
Example of SAprotocol.inp  file
#SA protocol
#nSAstep
2
#(f10.1,1x,f8.1,1x,3(f6.1,1x)
#234567890x12345678x123456x123456x123456
#ntimeMX    tempTg  SCvdW wfHb128BB wfhB128BS
100000      500.0     0.8    1.0     1.0
100000      100.0     1.0    1.0     1.0
END
#
   ntimeMX - number of md timeStep
   tempTg  - target temperature in K, this temperature will be reach during ntimeMX steps
   SCvdW   - parameter 0 - 1 to defile softness of the van der waals potential. Soft potential
             modifies Potential Energy Surface decrease a barriers of conformational transitions
   wfHb128BB, wfhB128BS - scaling factors for BackBone-BackBone and BackBone-SideChain Hydrogen Bond energy
```

moveRes file - moveRes file. User defined moving residue segments Default name ./moveRes.inp.

Output:

Result PDB - The final output PDB file with the best prediction. Other output PDB files will be stored in the result folder.

Result PDB - Run output file.

Molecule name - molecule name myMolec. The name will be added to the left of all files generated by the program, i.e. sequence of molecular dynamics trajectory snapshot files myMolec_mdResXXXX.pdb, molecular dynamic trajectory energy file myMolec_engMd.tra, the final result of mdynSB rum file myMolec_mdXYZVfin.pdb

MolDyn_Doc

Preference

mDynDock011 - Blind Hierarchical Docking of Ligand on Protein molecule

1) BLIND DOCKING OF FLEXIBLE LIGAND ON FLEXIBLE PROTEIN:
   a) ehaustive calculation of low-resolution binding sites;
   b) automatic blind docking of Ligand on Protein via hierarchical method including refinement of ligand position/orientation and conformation and protein bindin site conformation via

molecular dynamic simulated annealing coupled with force field deformation of ligand-protein interactions;

c) docking for user defined initial low-resolution sites on Protein

Note. Typical CPU time consumption for docking of small ligands is about 10 minutes per 1 binding site, and up to 1 hour for large ligands.

Installation and RUN

1.DOS/Linux installation

1)DOS installation
make dir myMdynDock/
put archive of program mMdynDock to this directory
extract archive,
subdirectories:

```
            myMdynDock011/dat   have to contain all  file.dat - Lib data for program
            myMdynDock011/src   fortran code files
```

MAKE enviromenr variable

```
> set MDYNDOCK011HOME="fullPath to myMdynDock011/dat"
```

myMdynDock011/doc directory contains Mannual and *.txt files
explaining how to run program and undestand results of calculations

RUN program from separate /jobX directory

2) Linux installation

```
  a) copy mDynDock011.tgz to myMdynDock/  directory
  b) extract archive
   > tar -zxvf  mDynDock011.tgz
  c) set env variable $MDYNDOCK011HOME
   > . ./setMDynDock011.sh
```

#- -

2. RUN program command

```
  $> $MDYNDOCK011HOME/mDynDock011 -i inProtcol  -c inProtPDB -cL inLigPDB
        -sa saProtocol [-mn molName]
        [-tL ligMolTopoDat] [-bsX bsXYZinPDBfile] -o runOutFile
        [-er errorFile]
```

in parenthesis [ ] are auxiliary files. The auxiliary files will be used by program if the main command file defines the respective task.

3. Command line DESCRIPTION:

RUN the MdynDock011 program by the command line

```
  $> $MDYNDOCK011HOME/mDynDock011 -i inProtcol  -c inProtPDB -cL inLigPDB
          -sa saProtocol [-mn molName]
          [-tL ligMolTopoDat] [-bsX bsXYZinPDBfile] -o runOutFile
          [-er errorFile]
```

in parenthesis [ ] are auxiliary files. The auxiliary files will be used by program if the main command file defines the respective task.

Command line file DESCRIPTION:

-c inProtPDB : file of the initial protein structure in the PDB format

-sa saProtocol : file defines simulated annealing protocol
-mn molName : character set defining prot-Lig name. molName will be attached as prefix
: to RESULT files
-cL inLigPDB : Ligand allAtom structure PDB file
-tL ligMolTopoDat : file of LigandMolecTopology created by the program mTopoHQ
-bsX bsXYZinPDBfile: file of low-resolution binding site positions to
be refined { can be taken from the result file
LigBSiteOnSAS01.pdb - XYZ of low-resolution
binding sites} see description in the file
P05-MdynDock011-TEST1-8-EXAMPLE.tx
-o runOutFile : run output file

Current status of program run is printed on the standard output device (consol) or
can be redirected to user defined file
-er errorFile : error message file : they are dublicated in the runOutFile
#
if file name definition in the argument line is missing for a file
than the default name is used for this file

NOTE! if the command line does not include a key -X , while the command file defines task
which needs an additional data file coupled with -X keyword,
than program try to find default (standard) name data file in the current directory.
#
Default names:
#
inProtcol = ./MdynDockPar.inp
inPDB = ./molec.pdb
saProtocol = ./SAprotocol.inp
molName = space
runOutFile = ./mDynDock.out
errorFile = ./mDynDock.err
#

inProtocol file description

The main command file {key -i inProtocol} for mDynDock011 program is the file
inProtcol = by default ./MdynDockPar.inp

The main command file consist of lines with command keyword.
Keyword start with $ sign in the first position of line
One Keyword in line

#example of MdynPar.inp file and keyword description

```
# MdynDockPar.inp
#
# DEFINITION of the DOCKING PROTOCOL
# Five docking protocols can be defined

1) DOCKING FOR one USER DEFINED position/orientation of LIGAND
$doLigDock=10       !run docking for [USER defined] initial position&orientation of Ligand
                    ! as it is in the initial inPDB file [united pdb file of protein + ligand]


$doLigDock=11           !run docking for [USER defined] initial
                  position of Ligand
                 ! as it is in the initial inPDB file [united
                   pdb file of protein + ligand]
                 ! Docking is done via simulated annealing molDynamics
                 ! coupled with temperature and force field variation.
                 ! Ligand CMass can move in vicinity of
                  initial
                 ! position +/- 4.0 A
                 ! Orientational global optimization are done
                  via
                 ! simulated annealing MD with multiple start
                 ! orientations. Initial orientations are uniformly
                 ! cover all orientational phase space with
                  step  = 90 deg  -
                 ! COARSE GRANE 24-point orientational grid

                 !run docking for [USER defined] initial
                  position of Ligand
$doLigDock=12           ! Initial orientations for Orientational
                  global optimization are uniformly
                 ! cover all orientational phase space with angle  = 45 deg
                 ! between two neigbour orientations - FINE
                  GRADE 144-point orientational grid
---------------------------------------------
2)EHAUSTIVE BLIND DOCKING
$doLigDock=21           ! run ehaustive blind docking for protein
                          molecule with coarse grane 24-point
                          orientational grid =24 orreintations uniformly distributed
                       over unit sphera, 90 degrees between two neighbor orientations

$doLigDock=22           ! run blind docking with fine grane
                        orientational grid =144 orreintations uniformly distributed
```

over unit sphera, 45 degrees
between two neighbour orientations

$doLigDock=23 !run blind docking whith medium grane
orientational grid
=72 orreintations uniformly distributed
over unit sphera, 60 degrees
between two neighbour orientations

! These option initiates seach of all binding sites on the
! protein molecular surface including
cavities and crevices via algorithm:.
! 1) search of surface cavities, crevicies
and groovs on protein surface
! 2) calculation and scoring of low-
resolution binding sites positions XYZ based on the number of ligand-
protein atom-atom contacts.
! 3) fine ligand docking by simulated annealing
molecular dynamics

DEFINE LIGAND & PROTEIN FLEXIBILITY

1. LIGAND is considered as fully flexible for all docking protocols.
PROTEIN can be considered as a **fixed** (rigid) structure or as having **flexible** region around the specified distant to Ligand atoms

------

Protein flexibility is defined by keyword
$flexBindSiteRad=**5.0** !flexProtein within 5.0 (or any real xx.x)
Angstrom from LigAtoms is a user defined value.
! the atoms of protein residue is considered as moving
(flexible) if the distance between any
atom of the residue and any atom of Ligand
is less than **5.0** A (user defined)

$OUTfull ! full extended output of program run
! by default OUTshort option is ON
#Initial PDB data quality
$Hread ! read INPUT pdb file with
Hydrogens – is the only default option for docking

#FORCE FIELD MODIFICATIONS:
#
$hBond128 = 2.0 ! scaling coeff for H-bonds
! default=1.0 is standart force field
#
$aSoftCore = 0.6 !invoke SOFTNES for the van der waals atom-atom potential

```
                           ! at the small (contact) atom-atom distances
                           ! for the energy optimization & MD equilibration run
                           ! Use of the softCore VDW potential helps to optimize
                           ! BAD molecular structures with many spartial atom-atom clashes
                           ! values range  0 - 1 from very Soft to standart VDW

#SOLVATION MODEL
$SolvMod = GShell
#
#
#OPIMIZATION PROTOCOL:
$engCalc                     ! do energy calculation
$engOptim                      ! do energy optimization by local Optimizer
$nOptStep=1                     ! max N optim steps
#
#PROTOCOL for Molecular Dynamic equilibration
$doMDyn                        !do MD equilibration&optimization

$MDSA                         !do MolecularDynamis
                          SimAnnealing
                          needs SAprotocolFile -sa
                          saProtocol File,
                          see additional description
#
#PROTOCOL of MD equilibration:
#
$initMDTemp=10.00               !initial Temperature to start
                          MolDyn
$bathMDTemp=50.00               !temperature of thermostat i.e.
                          target temperature
$runMDnstep=2000               !number of time-steps for MD
                          simulation
$mdTimeStep=0.002              ! time step for MD integrator
#
#
END
#
```

NOTE that parameter file formatted, i.e. $ sign should be the firs character of the line
----

Test examples

Test examples for all possible docking options:
There are 8 test examples for docking in the dir ./tLg

1) 1bty - benzamidine + trypsine complex
2) 1dwb - benzamidine + thrombin complex
3) 1dwc - alpha-Thrombin/MIT ligand complex
4) 1stp - biotin + streptavidine complex

5) 3tpi - ILE-VAL peptide + trypsinogen/BPTI complex
6) 1hvr - complex HIV-1 protease/XK263 ligand
7) 4phv - complex HIV-1 protease/VAC inhibitor
8) 1hiv - complex HIV-1 protease/NOA Ligand (119 atoms)

To run Ligand docking by respective script runMdynDock011.bpty.sh etc.

Each folder includes complete example of program usage with all command at auxiliary files.

```
----------------------------------------------------------------------------------------------
# Define DOCKING protocol:

$doLigDock=1             !run docking for [USER defined] initial
                   position of Ligand
                  ! as it is in the initial -cL inLigPDB file

                  ! Docking is done via simulated annealing molDynamics
                  ! coupled with temperature and force field
                   variation.
                  ! Ligand CMass can move in vicinity of
                   initial
                  ! position +/- 4.0 A
                  ! Orientational global optimization are
                   done via
                  ! simulated annealing MD with multiple
                   start
                  ! orientations. Initial orientations are
                   uniformly
                  ! cover all orientational phase space with
                   step  = 90 deg  -
                              ! COARSE GRANE 24-point orientational grid

                  !run docking for [USER defined] initial
                   position of Ligand
$doLigDock=12            ! Initial orientations for Orientational
                   global optimization are uniformly
                  ! cover all orientational phase space with angle  = 45 deg
                         ! between two neigbour orientations - FINE GRADE
               144-point orientational grid

#
$doLigDock=21        ! run ehaustive blind docking for protein
                molecule with coarse grane orientational grid
                   =24 orreintations uniformly distributed
                over unit sphera, 90degrees
                           between two neighbour orientations
$doLigDock=22        ! run blind docking with fine grane
                 orientational grid
                   =144 orreintations uniformly distributed
               over unit sphera, 45 degrees
                          between two neighbour orientations
```

```
$doLigDock=23         !run blind docking whith medium grane
                      orientational grid
                      =72 orreintations uniformly distributed over
                      unit sphera, 60 degrees
                      between two neighbour orientations

                        ! This option initiates  seach of all
                          binding sites on the
                        ! protein molecular surface including
                          cavities and crevices via algorithm:.
                        ! 1) search of surface cavities, crevicies
                            and groovs  on protein surface
                        ! 2) calculation and scoring of
                            low-resolution binding sites
                        !    positions based on the number of
                           ligand-protein atom-atom contacts.
                        ! 3) ligand docking by simulated annealing
                             molecular dynamics for best
                        !    candidate binding sites.
```

#REMARKS:

1) -c inPDBfile in command line should include proteinXYZ
   -cL inLigPDB is the ligandXYZ.

2) For a new Ligand, the Ligand molecular topology can BE included into the LIBrary topology file for LIGands in the /dat directory

bs_lig_all94.dat

!!!or the command line to rum program mDynDock011 have to include key -tL newLigMolTopoFile
The newLigMolTopoFile for a new Ligand can be calculated by the mTopoQ011 program

At the moment the topology file bs_lig_all94.dat includes 12 molecular Ligands:

1) benzamidine - BEN
2) biotine - BTN
3) agrotroban - AGT
4) agrotroban - MIT
5) VAC - inhibitor HIV1 protease, complex 4phv
6) KNI - inhibitor HIV1 protease, complex 1hpx
7) XK263 - inhibitor HIV-1 protease, complex 1hvr
8) MID - LIG from 1dwd complex, inhibitor HIV-1 protease
9) A77 : LIG from 1hvi complex, inhibitor HIV-1 protease
10) MIE : LIG from 1ett complex, epsilon-thrombin
11) DMQ : LIG from 1dmp complex, inhibitor HIV-1 protease
12) ICL : LIG from 1inc complex, elastase
#
Ligands of peptide nature, i.e. Ile-Val as it is in the test example, etc. can be run with available LIBrary of aminoasid residue topology data.
The aminoacid residues topology data files: bs_fin_all94.dat, bs_int_all94.dat, bs_one_all94.dat includes a list of unusual residues of peptide nature

for example:
N-end residue TFA, C-end residues ISO, ANI to build a topology of ligands of peptide nature, as in the complexes:
1ela Ligand: TFA-LYS-PRO-ISO
1elb : TFA-LYS-LEU-ISO
1elc : TFA-LYS-PHE-ISO
1eld : TFA-PHE-ALA-ANI
1ele : TFA-VAL-ALA-ANI
#
#
Another unusual residues in LIB files are
NLEH, NLE : complex 4hvp
PHEH, PHEC, OME : complex 1hef
PHEH, PHEC, CLYC, OME : complex 1heg
NOA, CAV, APY : complex 1hiv
#

```
Example of recomended main parameter file:
#
#MdynPar_1stp.inp for ligand Docking
#-----------------------------------------------
# 1stp : biotin - streptavidin complex
#234567890123456789012345678901234567890!comment
#$OUTfull

$flexBindSiteRad=5.0                           !flexProtein within 5.0 A from
LigAtoms
$doLigDock=21                            !do Lig blind Docking (2) on
                                          the protein
                                         ! using grade=1 rotational
                                          grid  (consist of 24
                                          uniformly
                               ! distributed rotational states)
                               ! another options are : 22 and 23
                                 consists of
                               ! 144 or 72 rotational grid  states

$hBond128=2.0                            !=scalingCoef for LibDatH128
$Hread
$SolvMod = GShell
$doMDyn
$MDSA                                    !do SimAnnealing
$aSoftCore=0.50                          !softCore 0->1
$initMDTemp=30.00
$bathMDTemp=50.0
$runMDnstep=1000
$mdTimeStep=0.002
$nwtra=1000
#END

SAprotocol_1stp.inp
# recomended Simulated annealing protocol file  for docking
# --------------------------------------------------------
#SA protocol_long_flex_protein
#234567890x12345678x123456x123456x123456
#       ntimeMX    tempTg  SCvdW rigidSC   rigidBB
# SCvdW - scaling factor for VDW interaction at small atom-atom distances
SAPROT  2000       100.0     0.1    1.0     1.0
SAPROT  2000       300.0     0.2    1.0     1.0
SAPROT  2000       300.0     0.4    1.0     1.0
SAPROT  2000       300.0     0.6    1.0     1.0
SAPROT  2000       200.0     0.6    1.0     1.0
SAPROT  2000       100.0     0.8    1.0     1.0
SAPROT  2000        50.0     0.9    1.0     1.0
SAPROT  4000        50.0     1.0    1.0     1.0
#flexible protein SideChain
#rigidSC 1.0/0.0 - rigid/flex SIDE chain of protein in a vicinity of BINDING
site
SAPROT  2000       200.0     1.0    0.0     1.0
SAPROT  2000        50.0     1.0    0.0     1.0
SAPROT  4000        50.0     1.0    0.0     1.0
#flexible protein SideChain + BackBone chain
#  rigidBB  -  1.0/0.0  rigid/flex  protein  BackBone  chain  in  a  vicinity  of
BINDING site
SAPROT  4000        50.0     1.0    0.0     0.0
END
```

!NOTE:

If the **tempTg** at a line **n** of the file SAprotocol.inp is EQUAL to the **tempTg** of the line **(n-1)** then simulated annealing at the n's step of protocol choose the snapshot with the minimal PotentialEnergy as the result of the n's step of the simulated annealing protocol – **recommended** to search the refined ligand pose.

##

UNDERSTANDIG the DOCKING RESULTs :

1bty - benzamidine + trypsine complex
/test011/bpty

2.1 File LigBSiteOnSAS01.pdb
LigBSiteOnSAS01.pdb - is the low-resolution ligand binding sites with ContactScore

```
 #LigBindGridOnSAS:            XYZ of low-resolution site   ContactScore
ATOM      1  LBSt        1      16.536  26.130   8.764   11
ATOM      2  LBSt        2      29.319  14.972  16.378   11
ATOM      3  LBSt        3       6.595  15.454  32.366    9
ATOM      4  LBSt        4      28.049  26.396   3.572    9
ATOM      5  LBSt        5      37.370  14.662  29.278    8
ATOM      6  LBSt        6       9.605  28.662  39.481    7
ATOM      7  LBSt        7      18.280  35.574  15.402    7
ATOM      8  LBSt        8      30.648  34.679  44.060    7
ATOM      9  LBSt        9      34.040  33.767  21.484    7
ATOM     10  LBSt       10       5.056  19.922  18.987    6
ATOM     11  LBSt       11      25.308   5.865  13.437    6
ATOM     12  LBSt       12      13.241  31.812  30.019    6
ATOM     13  LBSt       13       6.174  15.317  15.623    6
ATOM     14  LBSt       14      15.230  11.995  39.322    6
ATOM     15  LBSt       15      42.858  27.966  33.933    6
ATOM     16  LBSt       16      39.046  14.805   5.421    5
ATOM     17  LBSt       17      24.676  37.002  14.221    5
```

--

```
 2.2. LigDockFin_ePL.res file are the energy of protein/ligand interactions
      for high-resolution docking (all atom Ligand)
 #example: 1bty
NN    LigDockFinXXX.XXX.pdb        ePLtot   eVDW      eCoul     eHb       eSolv     eGeo
tempTAv
   1 ./LigDockFin001.001.pdb    -21.3    -9.6    -0.9    -9.5    -4.0      2.8      46.4
   2 ./LigDockFin001.002.pdb    -21.3   -10.5    -1.6    -9.2    -3.6      3.7     46.4
   3 ./LigDockFin001.003.pdb    -21.2    -9.4    -1.5    -9.8    -3.8     3.3    46.4
   4 ./LigDockFin002.001.pdb    -32.0   -20.5    -6.5   -12.9     2.9     5.0    51.0
#native Bind Pose
   5 ./LigDockFin002.002.pdb    -32.0   -21.5    -6.1   -12.5     2.8     5.4    51.0
   6 ./LigDockFin002.003.pdb    -31.8   -21.1    -6.0   -12.8     2.7     5.4    51.0
   7 ./LigDockFin003.001.pdb    -20.7    -2.2    -2.3    -9.8    -9.6     3.2    46.5
   8 ./LigDockFin003.002.pdb    -20.5    -2.2    -2.2    -9.5    -9.9     3.4    46.5
   9 ./LigDockFin003.003.pdb    -20.4    -1.9    -2.8    -9.6    -9.6     3.4    46.5
  10 ./LigDockFin004.001.pdb    -28.7    -4.4    -7.8    -9.9    -8.9     2.3    51.0
  11 ./LigDockFin004.002.pdb    -28.6    -6.2    -6.6    -9.8    -8.7     2.7    51.0
  12 ./LigDockFin004.003.pdb    -28.5    -5.3    -8.2    -9.8    -8.5     3.2    51.0
  13 ./LigDockFin005.001.pdb    -28.3   -11.4    -4.9    -9.8    -4.6     2.5    44.7
  14 ./LigDockFin005.002.pdb    -28.2   -11.6    -5.1    -9.7    -4.6     2.9    44.7
  15 ./LigDockFin005.003.pdb    -28.0   -11.5    -4.8    -9.8    -4.5     2.6    44.7
  16 ./LigDockFin006.001.pdb    -23.1    -3.5    -3.2   -13.0    -7.5     4.1    52.5
  17 ./LigDockFin006.002.pdb    -22.7    -4.0    -3.5   -12.8    -7.0     4.4    52.5
  18 ./LigDockFin006.003.pdb    -22.7    -4.0    -3.1   -13.2    -7.2     4.8    52.5
  19 ./LigDockFin007.001.pdb    -22.1    -8.8     0.2    -9.9    -6.5     3.0    49.8
  20 ./LigDockFin007.002.pdb    -21.7    -6.9    -0.3    -9.7    -7.0     2.3    49.8
  21 ./LigDockFin007.003.pdb    -21.7    -7.5    -0.4    -9.4    -7.4     3.0    49.8
  22 ./LigDockFin008.001.pdb    -23.7    -4.9    -2.3   -13.0    -8.0     4.4    48.7
  23 ./LigDockFin008.002.pdb    -23.6    -4.8    -2.5   -12.7    -8.4     4.8    48.7
  24 ./LigDockFin008.003.pdb    -23.5    -5.1    -2.6   -12.6    -8.0     4.7    48.7
  25 ./LigDockFin009.001.pdb    -21.6    -7.2    -0.8    -9.7    -7.1     3.1    52.9
  26 ./LigDockFin009.002.pdb    -21.2    -6.2    -0.5    -9.7    -7.7     3.0    52.9

#
```

2.3 A list of high-resolution ligand structures

LigDockFin001.001.pdb
LigDockFin001.002.pdb
LigDockFin001.003.pdb
LigDockFin002.001.pdb
LigDockFin002.002.pdb
LigDockFin002.003.pdb
.... etc.

LigDockFin005.002.pdb
LigDockFin005.003.pdb
LigDockFin006.001.pdb
LigDockFin006.002.pdb
LigDockFin006.003.pdb

Where, the first number 00x. is the number of **the low resolution binding site** the second number .001 .002 .003 are the best three orientations found by global optimization by the MD simulated annealing protocol.

The binding mode LigDockFin005.001.pdb has the minimal energy of Ligand interaction with protein this mode should taken as the best #1 in the blind docking procedure.

```
--------------------------------------------------------------------------------
#
Example of recomended main parameter file:
#
#MdynPar_1stp.inp for ligand Docking
#-----------------------------------------------
# 1stp : biotin - streptavidin complex
#234567890123456789012345678901234567890!comment
#$OUTfull

$flexBindSiteRad=5.0                            !flexProtein within 5.0 A from
LigAtoms
$doLigDock=21                             !do Lig blind Docking (2) on
                                            the protein
                                          ! using grade=1 rotational
                                            grid  (consist of 24
                                            uniformly
                                 ! distributed rotational states)
                                 ! another options are : 22 and 23
                                   consists of
```

```
                               ! 144 or 72 rotational grid  states
$hBond128=2.0                            !=scalingCoef for LibDatH128
$Hread
$SolvMod = GShell
$doMDyn
$MDSA                                    !do SimAnnealing
$aSoftCore=0.50                          !softCore 0->1
$initMDTemp=30.00
$bathMDTemp=50.0
$runMDnstep=1000
$mdTimeStep=0.002
$nwtra=1000
#END

#----------------------------------------------------------
#
SAprotocol_1stp.inp
# recomended Simulated annealing protocol file  for docking
# --------------------------------------------------------
#SA protocol_long_flex_protein
#234567890x12345678x123456x123456x123456
#       ntimeMX    tempTg  SCvdW rigidSC   rigidBB
# SCvdW - scaling factor for VDW interaction at small atom-atom distances
SAPROT  2000       100.0     0.1    1.0     1.0
SAPROT  2000       300.0     0.2    1.0     1.0
SAPROT  2000       300.0     0.4    1.0     1.0
SAPROT  2000       300.0     0.6    1.0     1.0
SAPROT  2000       200.0     0.6    1.0     1.0
SAPROT  2000       100.0     0.8    1.0     1.0
SAPROT  2000        50.0     0.9    1.0     1.0
SAPROT  4000        50.0     1.0    1.0     1.0
#flexible protein SideChain
#rigidSC 1.0/0.0 - rigid/flex SIDE chain of protein in a vicinity of BINDING
site
SAPROT  2000       200.0     1.0    0.0     1.0
SAPROT  2000        50.0     1.0    0.0     1.0
SAPROT  4000        50.0     1.0    0.0     1.0
#flexible protein SideChain + BackBone chain
# rigidBB - 1.0/0.0 rigid/flex protein BackBone chain in a vicinity of
BINDING site
SAPROT  4000        50.0     1.0    0.0     0.0
END
!NOTE:
```

If the **tempTg** at a line **n** of the file SAprotocol.inp is EQUAL to the **tempTg** of the line **(n-1)** then simulated annealing at the n's step of protocol choose the snapshot with the minimal PotentialEnergy as the result of the n's step of the simulated annealing protocol – **recommended** to search the refined ligand pose.

```
#--------------------------------------------------------
#
```

2) if $doLigDock=11, or (12, 13) , then docking of a ligand for User defined initial ligand position.

MD refinement of ligand position/orientation/conformation will be performed by MD with multiple initial orientation of the ligand. If doLigDock=11 then the set24 of the orientational grid (24-orientatons uniformly distributed in the orientational phase space). If doLigDock=12,13 then, the set160 or the set72 are used.

3) if $doLigDock=21, 22, 23, than a blind docking over whole protein surface is performed.

Docking with flexible protein residues in the vicinity of the Ligand can be performed by defining the keyword $
#
RESTRICTION:

A maximum size of flexible Ligand which can be docked via available method has been tested for a ligand size of < 200 atoms, e.g. eight-ten residue peptide.
#

Ligand Docking Method

Blind Docking method:
Exhaustive search of low-resolution binding sites and global optimization of ligand pose via MD simulated annealing coupled with force-field deformation
Method description:
Docking method is performed by subroutines **runLigDock01, runLigDock02 and runLigDock03** in the mDynDock011 program.
The program performs blind docking of molecular ligand of size up to ~200 atoms to a protein molecule.

The algorithm flow can be described as [1]:
1) Calculation of the accessible surface of the protein. Calculation of a surface grid for probe sphere of radius ~ average atomic radius, and contact positions [**bindSiteAt01(*)**]with protein atoms. Calculation are done by subroutine **surf_SAS04.**

2) Calculation of a surface grid points for a probe ligand of radius of typical aromatic ring [benzene] **gridsizeSAS** ~ 3.0 A. The surface grid are calculated by clustering of surface contact positions **bindSiteAt01(*)** and the surface grid **bindGridXYZSAS01(*)** is generated. The contact score [**nsasGridPoint(*)**] equal to the number of contact atomic positions included in to the surface grid point **bindGridXYZSAS01(*)** is calculated.
The **bindGridXYZSAS01(*)** are sorted by descent of the contact score value
nsasGridPoint(*) and presents an initial trial positions for refined docking of ligand.

3) Refined docking is performed via subroutine **runLigDock01**(ig,bindGridXYZSAS01loc)
For each initial positions **bindGridXYZSAS01(*)** for ligand center.
Procedure **runLigDock01** perform global optimization of ligand orientation and position in a restrained region of 3D-space. Spatial restraints are a sphere of radius equal to **gridsizeSAS.** Orientational optimization based on exhaustive search via optimization from different initial orientations uniformly covering all orientational space. The orientational optimization can be done in three modes. Coarse grain mode consist of 24 orientations with 90deg between two neighbor orientations, fine mode consist of 72 and 144 orientations with 60/45deg angle between two neighbor orientations. For each initial ligand orientation the molecular dynamic simulated annealing coupled with van der waals potential scaling is performed for flexible ligand and fixed protein atoms. A variant of deformable potential energy surface global optimization method is used. Three best final position/orientations of ligand are collected for each initial positions **bindGridXYZSAS01(*) in the files**
LigDockFinMMM.nnn.pdb – where MMM – grid position number, nnn – 001,002,003 – orientations.
The best docking variant for the ligand can be chosen as the file LigDockFinMMM.nnn.pdb with minimal potential energy engPOTENTLG: **.**

Test Examples:

1bty : benzamidine-trypsine complex

Table 1. Low-resolution binding sites

File	#LigBindGridOnSAS:			X	Y	Z	contactScore
ATOM	1	LBSt	1	16.536	26.130	8.764	11
ATOM	2	LBSt	2	29.319	14.972	16.378	11
ATOM	3	LBSt	3	6.595	15.454	32.366	9
ATOM	4	LBSt	4	28.049	26.396	3.572	9
ATOM	5	LBSt	5	37.370	14.662	29.278	8
ATOM	6	LBSt	6	9.605	28.662	39.481	7
ATOM	7	LBSt	7	18.280	35.574	15.402	7
ATOM	8	LBSt	8	30.648	34.679	44.060	7
ATOM	9	LBSt	9	34.040	33.767	21.484	7
ATOM	10	LBSt	10	5.056	19.922	18.987	6
ATOM	11	LBSt	11	25.308	5.865	13.437	6
ATOM	12	LBSt	12	13.241	31.812	30.019	6
ATOM	13	LBSt	13	6.174	15.317	15.623	6
ATOM	14	LBSt	14	15.230	11.995	39.322	6
ATOM	15	LBSt	15	42.858	27.966	33.933	6
ATOM	16	LBSt	16	39.046	14.805	5.421	5
ATOM	17	LBSt	17	24.676	37.002	14.221	5
ATOM	18	LBSt	18	39.100	25.116	6.122	5
ATOM	19	LBSt	19	25.156	6.498	5.813	5
ATOM	20	LBSt	20	14.736	13.757	2.279	5
ATOM	21	LBSt	21	35.933	31.703	11.547	5
ATOM	22	LBSt	22	45.035	21.844	22.099	5
ATOM	23	LBSt	23	12.210	8.874	28.161	5
ATOM	24	LBSt	24	11.197	11.080	32.573	5
ATOM	25	LBSt	25	25.549	16.554	-0.897	4
ATOM	26	LBSt	26	34.793	8.348	15.236	4
ATOM	27	LBSt	27	26.857	9.202	21.336	4
ATOM	28	LBSt	28	34.072	12.246	27.335	4

…..

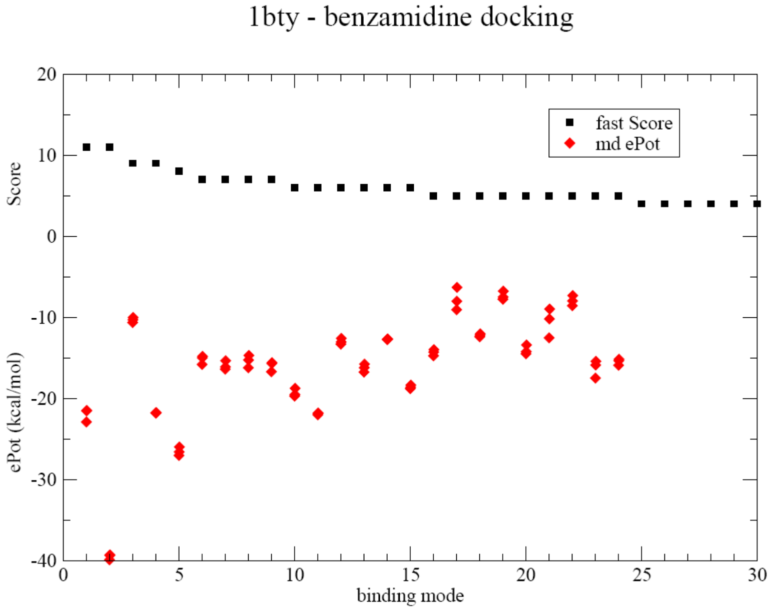
Table 2. Energy of binding for Refined binding modes of low-resolution binding sites.

NN	LigDockFinXXX.XXX.pdb	ePLtot	eVDW	eCoul	eHb	eSolv	eGeo	T,K	
1	./LigDockFin001.001.pdb	-21.1	-10.2	-1.0	-9.3	-3.8	3.1	51.6	
2	./LigDockFin001.002.pdb	-21.1	-10.5	-1.0	-9.5	-3.7	3.6	51.6	
3	./LigDockFin001.003.pdb	-21.0	-10.0	-1.3	-9.5	-3.4	3.3	51.6	
4	./LigDockFin002.001.pdb	**-32.1**	**-20.8**	**-5.9**	**-12.2**	**2.9**	**3.8**	48.9	*#bestDocking mode*
5	./LigDockFin002.002.pdb	-32.1	-21.3	-6.1	-12.1	2.8	4.7	48.9	
6	./LigDockFin002.003.pdb	-32.0	-20.6	-6.3	-13.2	2.8	5.2	48.9	
7	./LigDockFin003.001.pdb	-20.6	-2.1	-2.2	-9.7	-9.8	3.2	48.6	
8	./LigDockFin003.002.pdb	-20.4	-2.0	-2.5	-9.6	-9.8	3.6	48.6	
9	./LigDockFin003.003.pdb	-20.4	-2.3	-2.0	-9.4	-9.7	3.1	48.6	
10	./LigDockFin004.001.pdb	-28.7	-4.9	-7.6	-10.0	-8.5	2.3	47.7	
11	./LigDockFin004.002.pdb	-28.6	-5.0	-6.9	-9.9	-8.8	2.0	47.7	
12	./LigDockFin004.003.pdb	-28.5	-6.0	-7.1	-9.8	- 8.7	3.1	47.7	
13	./LigDockFin005.001.pdb	-28.5	-11.7	-4.5	-9.6	-4.9	2.3	50.2	
14	./LigDockFin005.002.pdb	-28.1	-11.2	-4.9	-9.8	-4.8	2.6	50.2	
15	./LigDockFin005.003.pdb	-28.1	-11.6	-5.2	-9.5	-4.6	2.9	50.2	
16	./LigDockFin006.001.pdb	-23.0	-4.0	-3.1	-12.9	-7.2	4.2	49.9	
17	./LigDockFin006.002.pdb	-23.0	-3.3	-3.6	-12.7	-7.6	4.1	49.9	
18	./LigDockFin006.003.pdb	-22.9	-4.3	-3.5	-12.4	-7.0	4.4	49.9	

1) 1bty complex benzamidine on trypsine

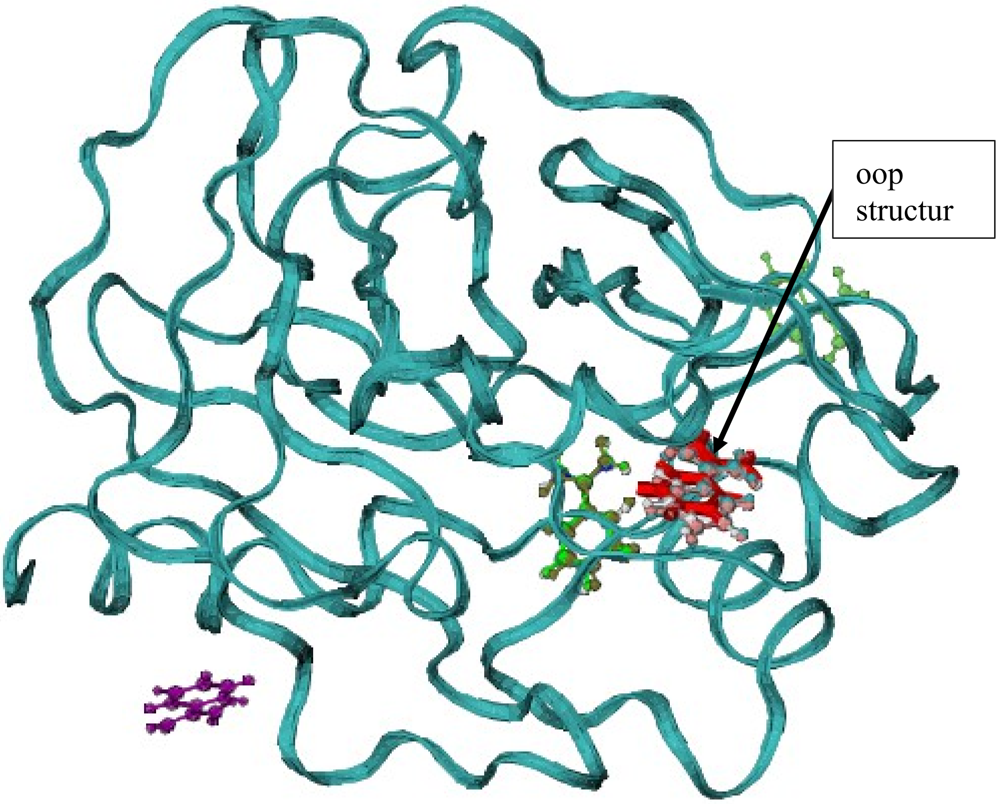
Fig.1 Docking results for benzamidine on trypsine – 1bty complex.

A – contact Score (black square) for binding grid points vs refined potential energy of ligand binding (red diamonds).



B – minimum energy docking mode (red bonds), RMSD = 0.54 A for all non Hydrogen atoms ligand of the native binding mode.

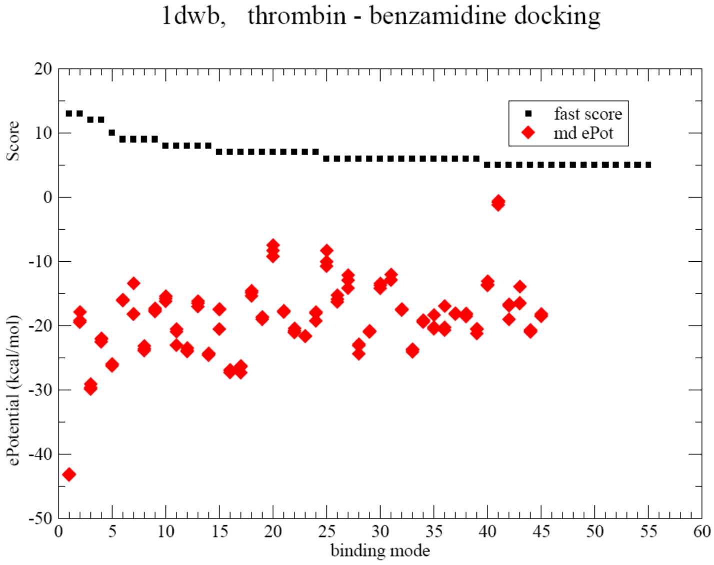
CPK- green and violet are less favorable binding modes with low binding energy are shown in (**A**). CPK (pink) - native binding mode of benzamidine in 1bty;



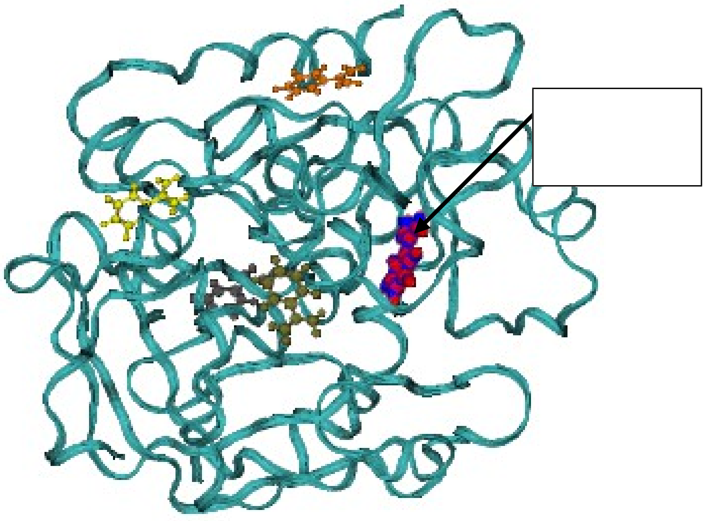
2) 1dwb : thrombin + benzamidine complex

Fig.2 Docking results for benzamidine on thrombin.

A : Contact Score (black square) for binding grid points vs refined potential energy of ligand binding (red diamonds).



B: (CPK blue)-minimum energy docking mode. Less favorable binding modes are shown – yellow, brown, green. CPK- (red) native benzamidine binding mode in 1dwb complex,

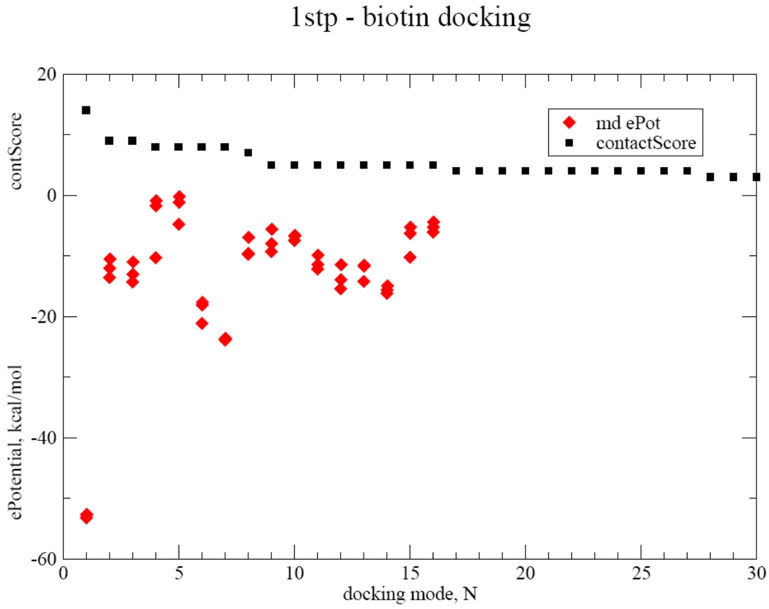


Minimum energy mode has RMSD = 0.27 A from the native binding mode of benzamidine.

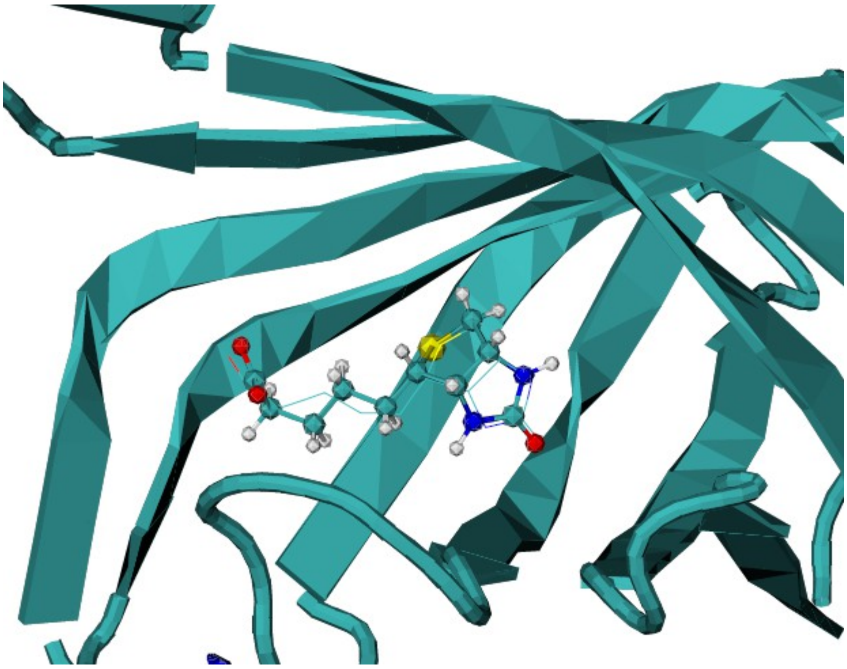
3) Biotine – streptavidine complex – 1stp

Fig.3. Docking result for biotine on streptavidine , 1stp complex.

A – contact Score (black square) for binding grid points vs refined potential energy of ligand binding (red diamonds).



B – minimum energy docking mode structure of biotine – CPK, lines – native biotine in the 1stp complex..

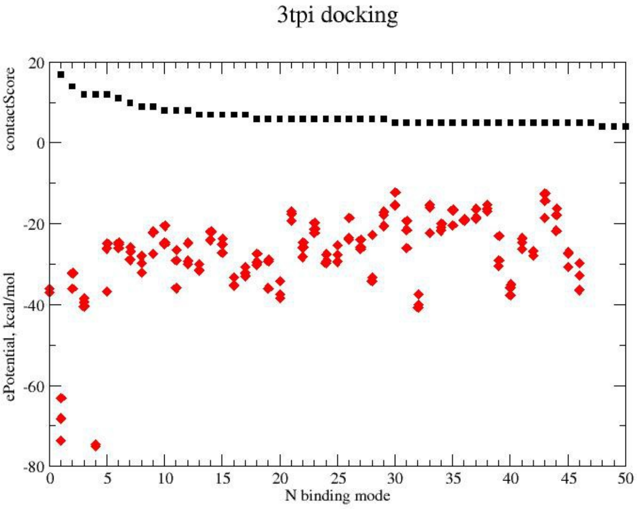


Minimum energy mode has RMSD = 0.96 A from the native binding mode of biotine.

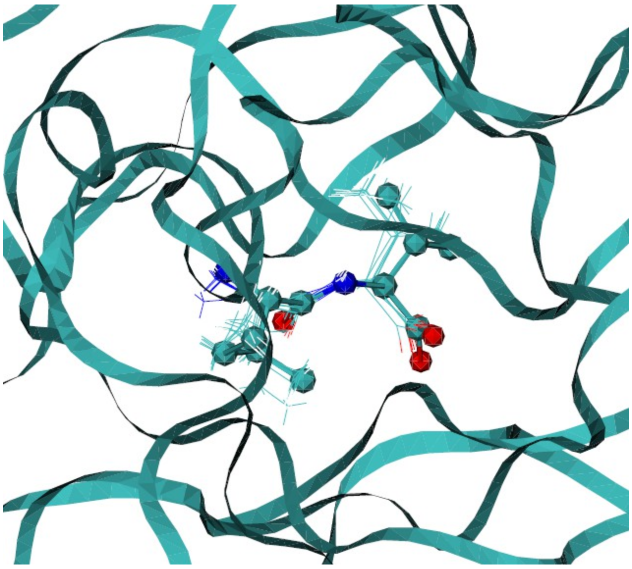
4) Trypsinogen/pancreatic trypsin inhibitor + Ile-Val peptide complex : 3tpi

Fig.4. Docking result for ILE-VAL dipeptide on **Trypsinogen/pancreatic trypsin inhibitor**.

A – contact Score (black square) for binding grid points vs refined potential energy of ligand binding (red diamonds).



B- Lines are minimum energy docking modes of rank 1- 4 structures of ILE-VAL peptide – lines, CPK – native binding mode of biotine in the 1stp complex..



The best binding energy mode has RMSD = 0.46 A from the native binding mode of dipeptide ILE-VAL

Table 1. Energies of top ranked binding modes, and RMSD from the native binding mode of the 3tpi.

Binding mode	ePL, kcal/mole	RMSD (A)
rank 1 - `LigDockFin001.001.pdb`	`-76.07`	0.46
rank2 - `LigDockFin001.002.pdb`	-75.6	0.58
Rank3 - `LigDockFin001.002.pdb`	-75.5	0.78
Rank4 - `LigDockFin004.001.pdb`	-74.8	0.88

5) 1dwc complex of Human thrombin with thrombin-inhibitor MIT .

Human thrombin – 296 residues

MIT – molecule includes 80 atoms

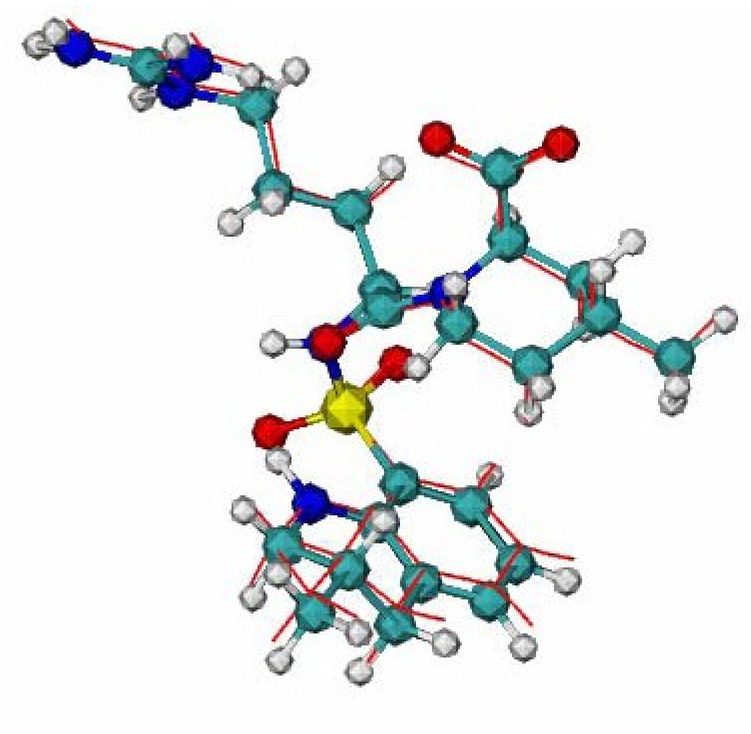


Fig. 1 Top Ranked calculated docking mode – red lines, CPK – native MIT in the native binding mode, RMSD = 0.2 A for calculated docking mode from the native.

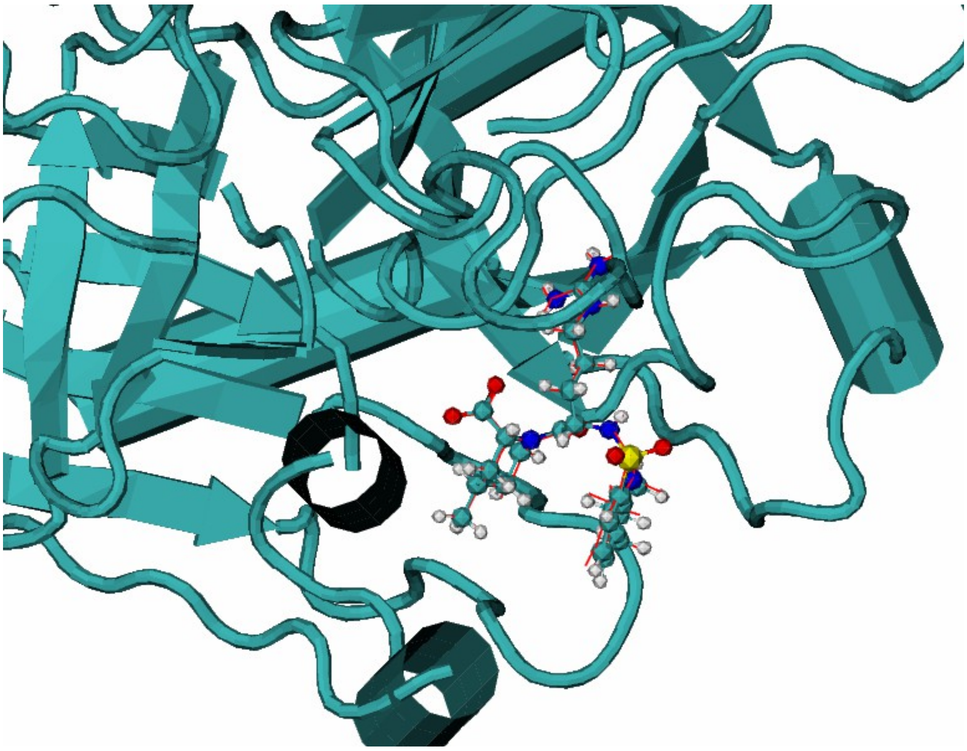


Fig.2. 1dwc complex. Red lines is docked MIT ligand, CPK is the native mode.

6) 1hiv complex of HIV1 protease with inhibitor NOA

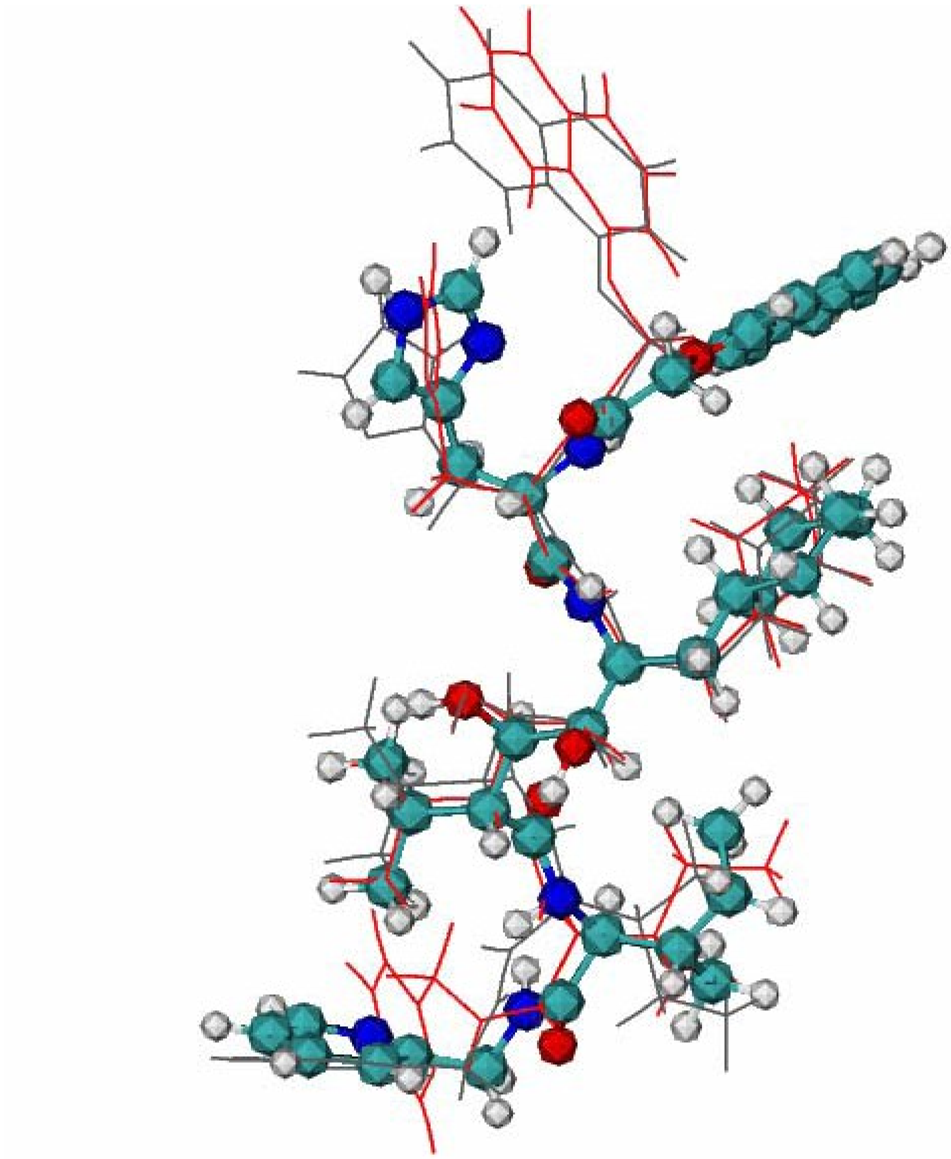
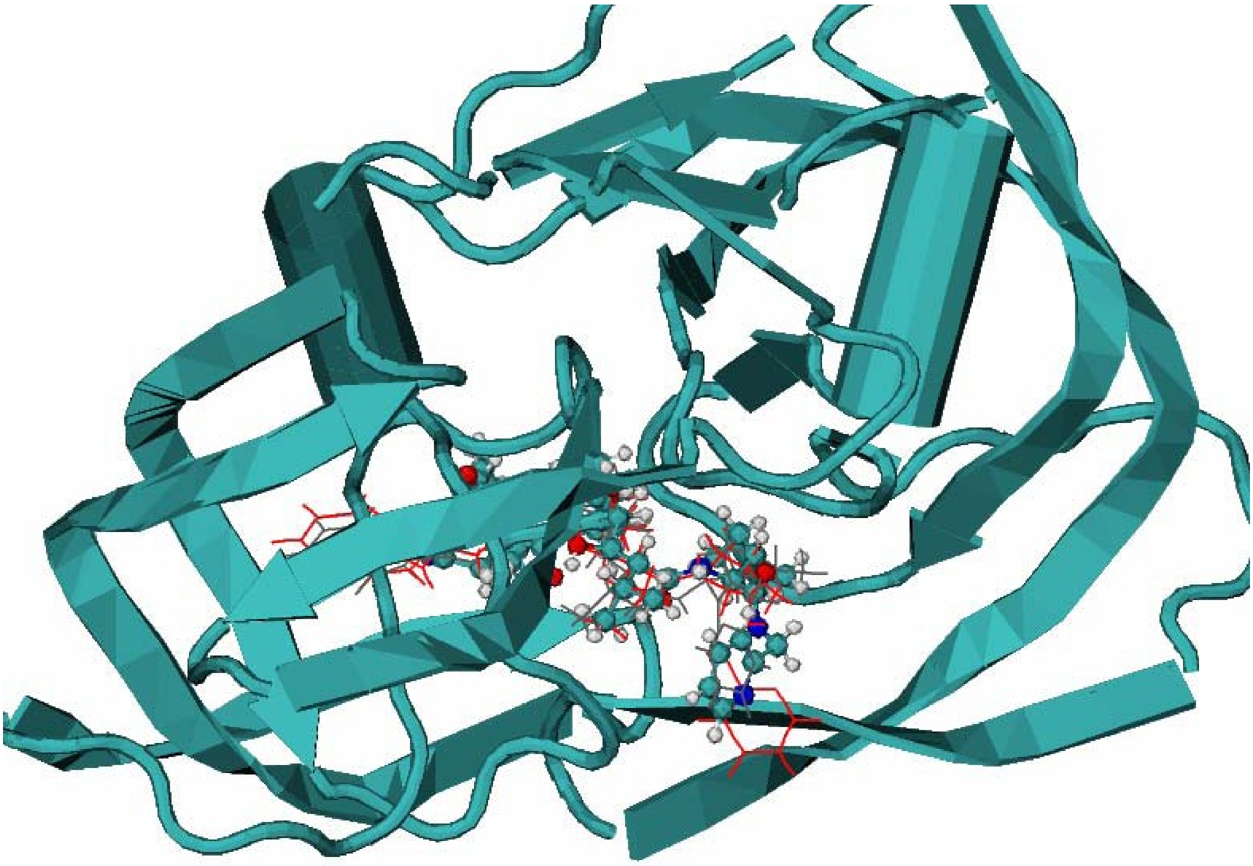


Fig.1. Two top ranked calculated binding modes of NOA in comparison with the NOA ligand in the native binding mode of 1hiv complex. CPK – native binding mode, lines (red and grey) the top ranked mode by energy of binding. The RMSD from the native are ~3.1A for all atoms. The major difference between native and calculated modes are the orientation of one aromatic double-ring at the top of molecule NOA, the RMSD = 1.1. A over all atoms except the later aromatic system.

Fig.2 . 1hiv complex of HIV1 protease with inhibitor NOA
CPK – native mode, red and grey lines – are calculated modes.



7) 1hvr complex of HIV1 protease with inhibitor XK2

Fig.1 Calculated binding mode of XK2, red lines, CPK – native binding mode of XK2 ligand. RMSD = 0.95 A for all atom.

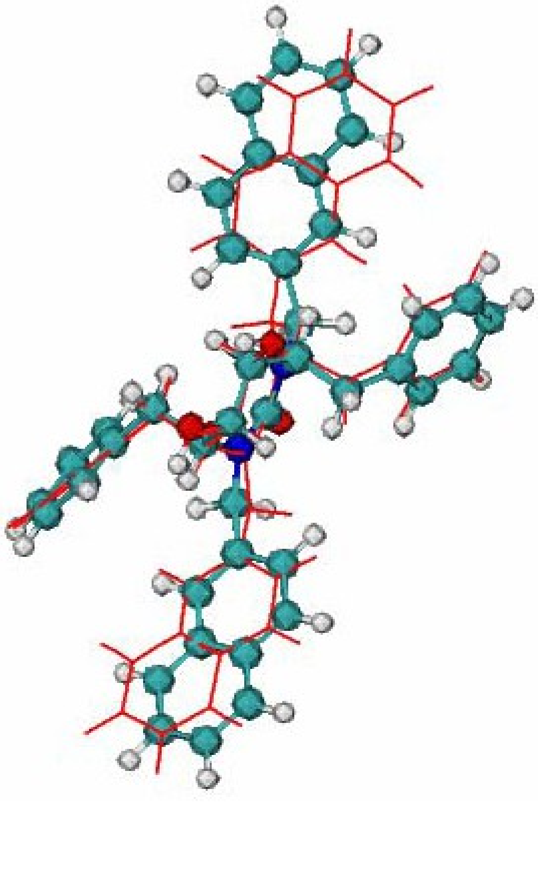
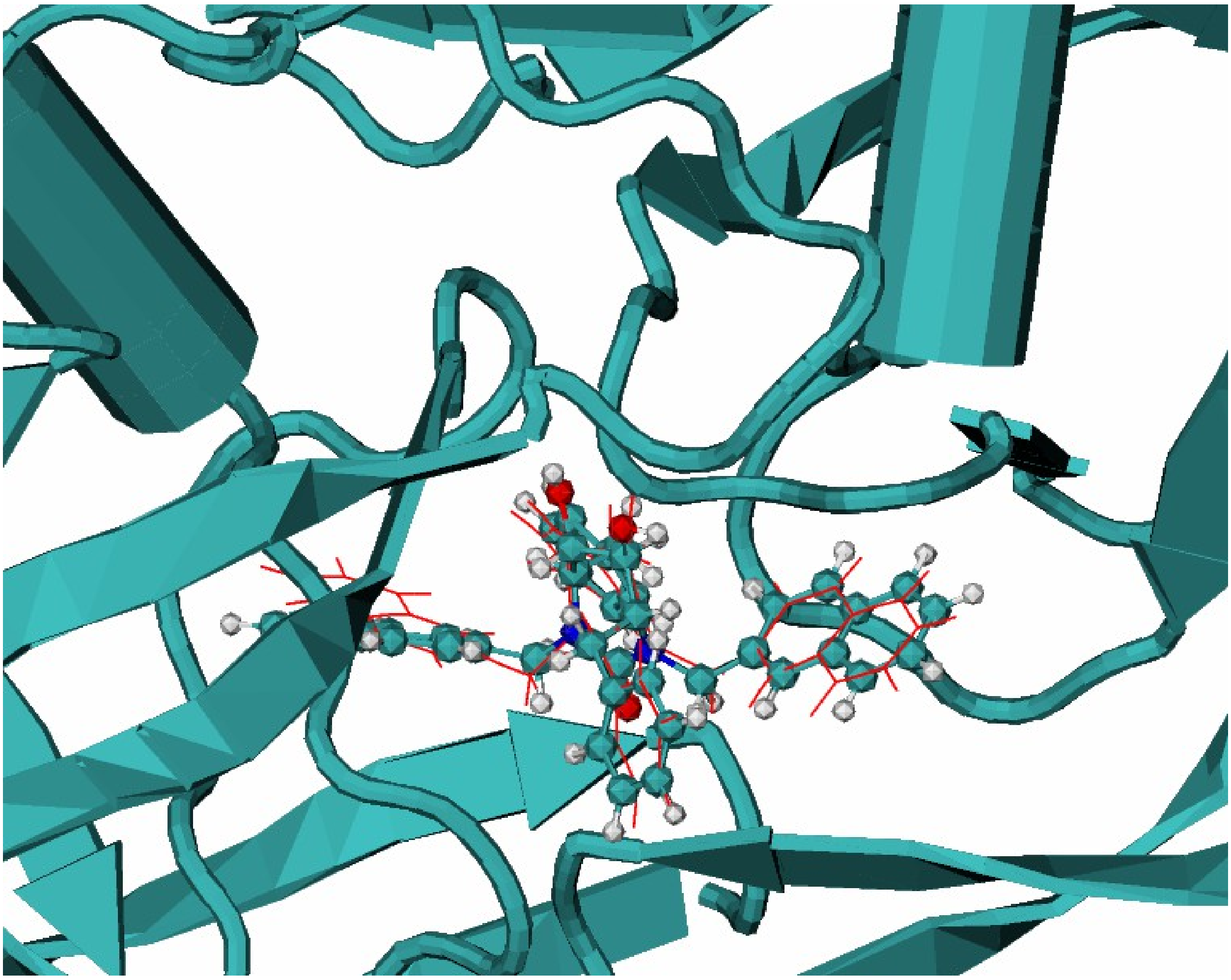


Fig.2 Calculated docking mode for the ligand XK2 in complex with HIV1 protease, CPK – the native binding mode of the XK2 ligand.



1hvp complex of 1HIV protease with VAC molecule inhibitor

Fig.1 Calculated best binding mode of VAC is in red lines, CPK – native VAC inhibitor in the 1hvp complex; the RMSD = 0.99 A.

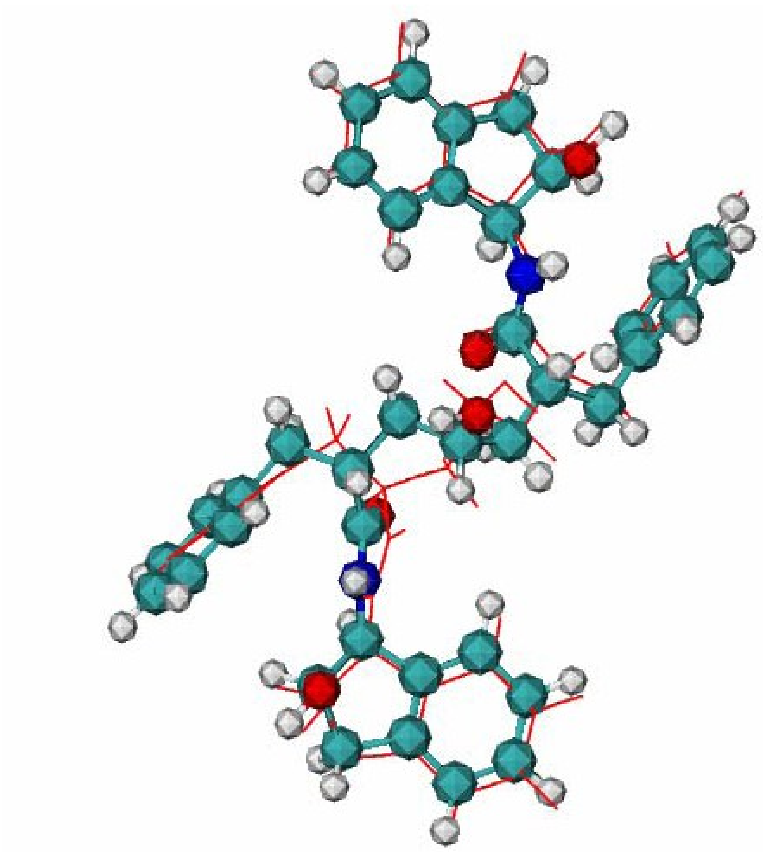


Fig.2. 4hvp complex, red lines is the calculated mode, CPK – the native binding mode of VAC inhibitor.

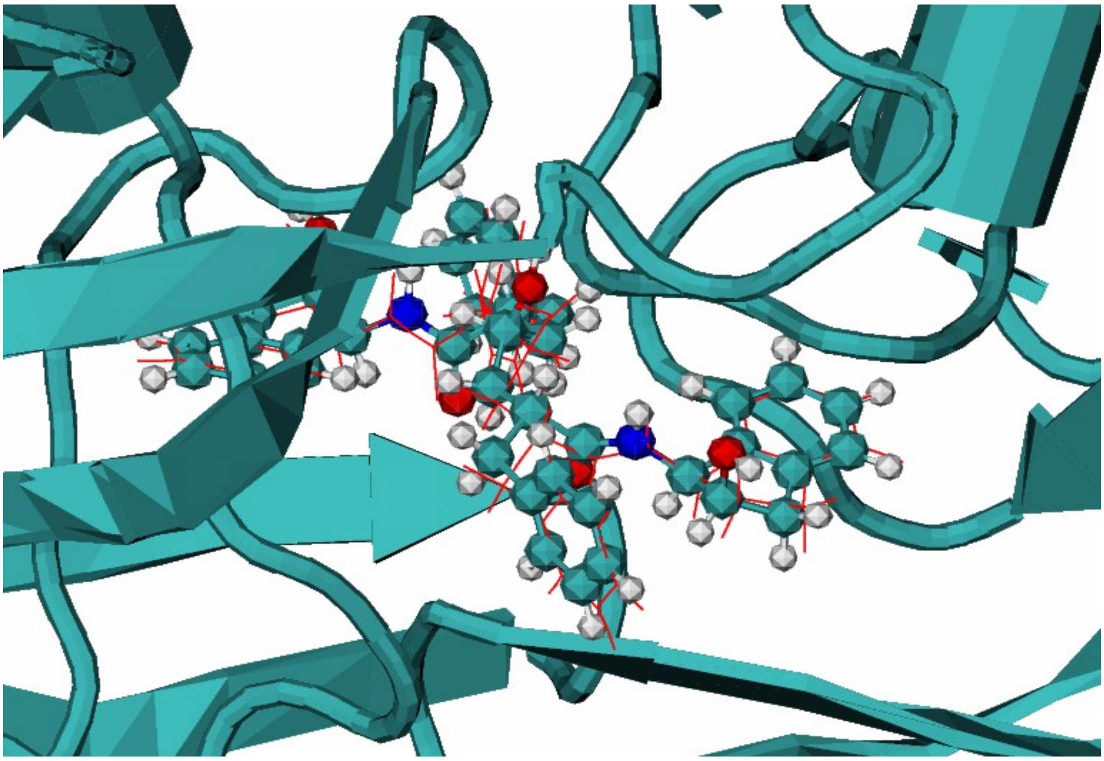


Table 1. Results of MdDock method for a set of complexes

complex	Ntors	RMSD, A	ΔEgap
1) 1bty trypsin/benz	0	0.5	3.7
2) 1dwb α-thrombin/benz	0	0.5	3.3
3) 1stp streptavidine/biotin	5	0.9	10.5
4) 3tpi trypsinogen/Ile-Vla	6	0.4	10.6
5) 1dwc α-thrombin/MIT	8	0.2	4.3
6) 1hiv HIV1 protease/NOA	16	1.1/3.1	3.6
7) 1hvr HIV1 protease/XK263	8	0.9	19.1
8) 4phv HIV1 protease/VAC	15	0.9/2.3	3.4

Ntors – number of flexible torsion angles.
ΔEgap – energy gap between lowest energy binding mod and the next energy mode**.**

Conclusion:
The developed method of blind docking show a good accuracy in prediction of the native bindig modes of flexible ligands. At the test set of 8 ligands the method shows 100% accuracy, i.e. the native binding mode are found as the mode with highest binding energy calculated in the all-atom force field. At a large set of protein-ligand complexes the hierachical blind docking method shows a high success rate of docking ~ 0.9 (see Vorobjev 2010) and outperforms some wide-used docking programs like AutoDock04.

Reference

1. Vorobjev Y.N. Blind Docking method combining search of Low-resolution binding sites
2. with ligand pose refinement by molecular dynamic based global
3. optimization. J. Comput Chem **31**, 1080-1090, (2010).

Examples of Ligand Docking Job on proteins

Directory ./tLg shows examples of Ligand Docking Job on proteins:

```
# DOCK TEST#1
```

./1bty - trypsine-benzamidine complex

1bty.ben.Native.pdb - protein-lig complex with NATIVE binding mode for Ligand

1bty.ben.notNative.pdb - protein-lig complex with notNATIVE (arbitryry) mode for Ligand

the both files can be used as inPDB file

```
#
#Run the mDynDoc program:
> . runMdynDock011.bpty.sh
```

or by the command line

```
> $MDYNDOCK011HOME/mDynDock011 -c 1bty.P.pdb -cL ben.notNative.pdb -i
MdynPar_1bty.inp -sa SAprotocol_long.inp
  -mn 1bty -o 1bty.out
#
```

UNDESTANDING DOCKING RESULTS:

1) file 1bty.bSiteAtOnSAS00.pdb shows positions of binding site candidates
   on the protein surface

```
 #LigBindGridOnSAS:                    XYZ                  SCore
 ATOM     1  LBSt      1     16.536  26.130   8.764   11
 ATOM     2  LBSt      2     29.319  14.972  16.378   11
 ATOM     3  LBSt      3      6.595  15.454  32.366    9
 ATOM     4  LBSt      4     28.049  26.396   3.572    9
 ATOM     5  LBSt      5     37.370  14.662  29.278    8
 ATOM     6  LBSt      6      9.605  28.662  39.481    7
 ATOM     7  LBSt      7     18.280  35.574  15.402    7
 ATOM     8  LBSt      8     30.648  34.679  44.060    7
 ATOM     9  LBSt      9     34.040  33.767  21.484    7
 ATOM    10  LBSt     10      5.056  19.922  18.987    6
 ATOM    11  LBSt     11     25.308   5.865  13.437    6
 ATOM    12  LBSt     12     13.241  31.812  30.019    6
   ...
  ATOM     40  LBSt      40     25.260   6.929  29.909    4
  ATOM     41  LBSt      41     26.781  13.047  43.008    4
```

Docking alhorithm put ligand center into this positions with SCore >= 6

and refine ligand orientation and conformation via semiglobal optimization by Simulated annealing coupled with protein-Ligand force field deformation.

#

The resulting ligand positions are collected in the files :

1bty.LigDockFin000.001.pdb

1bty.LigDockFin000.002.pdb

1bty.LigDockFin000.003.pdb

1bty.LigDockFin001.001.pdb

1bty.LigDockFin001.002.pdb

1bty.LigDockFin001.003.pdb

...

1bty.LigDockFin015.003.pdb

#

File 1bty.LigDockFin.ePL.res :

collects the refined total Potential energy of internal Lig-Lig + Lig-Prot

interactions for final Ligand Docking modes from

files 1bty.LigDockFin*.pdb

#

```
NN   LigDockFinXXX.XXX.pdb       ePLtot  eVDW    eCoul   eHb     eSolv   eGeo    tempTAv
   1 ./LigDockFin001.001.pdb     -24.3   -11.6    -0.6    -9.4    -6.9     4.3    50.2
   2 ./LigDockFin001.002.pdb     -24.2   -11.0    -1.2    -9.8    -6.8     4.6    50.2
   3 ./LigDockFin001.003.pdb     -24.0   -10.7    -1.3    -9.6    -6.9     4.5    50.2
   4 ./LigDockFin002.001.pdb     -39.3   -20.7    -6.2   -11.0    -4.5     3.1    51.8
- best ePLtot & RMSD mode
   5 ./LigDockFin002.002.pdb     -39.3   -20.6    -6.1   -12.8    -4.5     4.8    51.8
   6 ./LigDockFin002.003.pdb     -39.1   -20.7    -7.0   -10.3    -4.5     3.4    51.8
   7 ./LigDockFin003.001.pdb     -18.6    -1.2    -2.5    -9.4    -6.8     1.3    46.5
   8 ./LigDockFin003.002.pdb     -17.7    -2.4    -1.6    -9.6    -6.5     2.5    46.5
   9 ./LigDockFin003.003.pdb     -17.7    -2.1    -2.7    -9.7    -6.7     3.6    46.5
  10 ./LigDockFin004.001.pdb     -27.8    -4.5    -7.6    -9.9    -7.5     1.7    51.0
  11 ./LigDockFin004.002.pdb     -27.7    -5.6    -7.5    -9.6    -7.4     2.4    51.0
  12 ./LigDockFin004.003.pdb     -27.7    -5.4    -7.3    -9.9    -7.4     2.3    51.0
  13 ./LigDockFin005.001.pdb     -30.1   -10.8    -5.6    -9.6    -6.3     2.2    48.5
  14 ./LigDockFin005.002.pdb     -29.8   -11.6    -5.2    -9.5    -6.3     2.8    48.5
  15 ./LigDockFin005.003.pdb     -29.6   -12.4    -4.5    -9.5    -6.3     3.1    48.5
  16 ./LigDockFin006.001.pdb     -22.7    -9.0    -1.5    -9.5    -6.3     3.5    50.4
  17 ./LigDockFin006.002.pdb     -22.5    -8.6    -1.6    -9.4    -6.4     3.5    50.4
  18 ./LigDockFin006.003.pdb     -22.5    -7.9    -2.1    -9.3    -6.1     3.0    50.4
#
```

****

#

DOCK TEST#2

alpha-thrombin/bemzamidine complex : 1dwb

#

#Run the mDynDock program: by script

```
> . runMdynDock011.1dwb.sh
```

or by the command line

```
> $MDYNDOCK011HOME/mDynDock011 -c 1dwb.P.pdb -cL ben.notNative.pdb -i
MdynPar_1dwb.inp -sa SAprotocol_long.inp
 -mn 1dwb -o 1dwb.out
```

#

file 1dwb.LigDockFin.ePL.res: - total potential energy of Lig-Lig + Lig-Prot interactions:

#

```
NN   LigDockFinXXX.XXX.pdb    ePLtot  eVDW    eCoul   eHb     eSolv   eGeo  Tav  #rank
   1 1dwb.LigDockFin001.001.  -43.7   -16.6   -11.6   -13.9    -5.5     3.9  47.4 #1
   2 1dwb.LigDockFin001.002.  -43.3   -16.5   -11.0   -14.3    -5.5     4.0  47.4 #2
   3 1dwb.LigDockFin001.003.  -43.3   -17.1   -10.7   -14.1    -5.5     4.1  47.4 #3
   4 1dwb.LigDockFin002.001.  -25.2    -8.2    -2.9    -9.7    -7.7     3.4  48.4
   5 1dwb.LigDockFin002.002.    -24.9    -8.3    -3.4    -9.6    -7.7  4.0  48.4
   6 1dwb.LigDockFin002.003.    -24.6    -7.4    -3.4    -9.8    -7.6  3.6  48.4
   7 1dwb.LigDockFin003.001.    -31.6   -10.7    -6.1   -10.9    -7.1  3.2  50.1
   8 1dwb.LigDockFin003.002.    -31.2    -9.9    -6.3   -11.1    -6.9  2.9  50.1
   9 1dwb.LigDockFin003.003.    -31.2   -11.1    -5.8   -10.4    -7.0  3.1  50.1
  10 1dwb.LigDockFin004.001.    -20.2    -9.0    -2.1    -7.9    -5.6  4.4  50.4
  11 1dwb.LigDockFin004.002.    -19.7    -8.3    -2.2    -8.1    -5.7  4.5  50.4
  12 1dwb.LigDockFin004.003.    -19.4    -9.0    -2.3    -8.0    -5.7  5.5  50.4
  13 1dwb.LigDockFin005.001.    -29.9    -4.5    -9.4    -9.9    -8.3  2.2  52.1
  14 1dwb.LigDockFin005.002.    -29.9    -8.9   -10.7   -11.8    -9.0  10.5 52.1
  15 1dwb.LigDockFin005.003.    -29.8   -10.7    -6.4    -7.1    -9.0  3.4  52.1
  16 1dwb.LigDockFin006.001.    -21.9    -5.9    -2.0    -9.4    -7.3  2.7  51.3
  17 1dwb.LigDockFin006.002.    -21.8    -7.3    -1.7    -9.1    -7.2  3.5  51.3
  18 1dwb.LigDockFin006.003.    -21.7    -6.3    -2.4    -9.5    -7.2  3.6  51.3
  19 1dwb.LigDockFin007.001.    -23.7    -8.4    -1.2    -9.5    -7.8  3.2  49.4
  20 1dwb.LigDockFin007.002.    -23.6    -8.2    -1.8    -9.2    -7.8  3.3  49.4
  21 1dwb.LigDockFin007.003.    -23.1    -7.9    -2.1    -9.5    -7.6  4.0  49.4
```

****
****

#DOCK TEST#3

./1dwc   - alpha-Thrombin/MIT ligand complex

#

#Run the mDynDock program:

```
>  . runMdynDock011.1dwc.sh
```

or by the command line

```
> $MDYNDOCKHOME/mDynDock011 -c 1dwc.P.pdb -cL 1dwc.Lig.pdb -i MdynPar_d21.inp
-sa SAprotocol_long.inp -mn 1dwc -o
 1dwc.out
#
Docking result potential energy file:
1dwc.ePL.LigDockFin.res:
#
NN    LigDockFinXXX.XXX.pdb         ePLtot   eVDW      eCoul     eHb       eSolv     eGeo
tempTAv
   1 1dwc.LigDockFin001.001.    -64.5   -45.0   -14.3   -20.9   -17.5    33.3    49.7
- best ePL_mode
   2 1dwc.LigDockFin001.002.    -61.2   -42.9   -16.0   -22.3   -14.7    34.7    49.7
- best rmsd_mode
   3 1dwc.LigDockFin001.003.    -52.8   -33.9   -15.4   -22.3   -24.3    43.0    49.7
   4 1dwc.LigDockFin002.001.    -53.7   -35.1   -10.4   -24.2   -23.2    39.3    50.7
   5 1dwc.LigDockFin002.002.    -49.9   -30.1   -16.1   -21.7   -23.6    41.7    50.7
   6 1dwc.LigDockFin002.003.    -44.6   -29.8   -10.3   -22.1   -21.5    39.1    50.7
   7 1dwc.LigDockFin003.001.    338.3   194.2     5.7    -4.2   -13.8   156.4    53.1
   8 1dwc.LigDockFin003.002.    340.4   230.8   -12.9   -19.3   -15.6   157.4    53.1
   9 1dwc.LigDockFin003.003.    368.8   224.5    -2.5   -10.8   -13.3   170.9    53.1
  10 1dwc.LigDockFin004.001.    -49.8   -34.1   -13.2   -24.9   -14.4    36.8    50.5
  11 1dwc.LigDockFin004.002.    -48.6   -26.5   -13.2   -27.2   -14.5    32.8    50.5
  12 1dwc.LigDockFin004.003.    -47.2   -27.0   -16.5   -27.6   -12.1    36.1    50.5
  13 1dwc.LigDockFin005.001.    165.1    81.5    -2.2   -13.8   -19.8   119.4    52.8
  14 1dwc.LigDockFin005.002.    290.1   183.9    -8.2   -13.1   -11.4   138.8    52.8
  15 1dwc.LigDockFin005.003.    303.4   147.8     2.2    -8.3   -17.0   178.7    52.8
  16 1dwc.LigDockFin006.001.    -58.5   -37.5   -15.3   -18.4   -25.9    38.6    47.6
  17 1dwc.LigDockFin006.002.    -58.4   -37.1   -17.6   -25.7   -21.6    43.6    47.6
  18 1dwc.LigDockFin006.003.    -57.2   -30.6   -16.8   -18.6   -26.7    35.6    47.6
  19 1dwc.LigDockFin007.001.    -58.9   -26.3   -26.3   -29.3   -19.0    42.0    49.0
  20 1dwc.LigDockFin007.002.    -57.6   -26.6   -21.5   -24.0   -20.4    35.0    49.0
  21 1dwc.LigDockFin007.003.    -57.3   -28.2   -21.7   -24.5   -19.3    36.3    49.0
...
************************************************************************
***********
#
DOCK TEST#4
./1stp : complex streptavidine/biotin
```

#

#Run the mDynDoc program:

by script

```
> . runMdynDock011.1stp.sh
```

or by the command line

```
> $MDYNDOCK011HOME/mDynDock011 -c 1stp.P.pdb -cL 1stp.btn.Lig.pdb -i MdynPar_d21.inp -sa SAprotocol_long.inp -mn 1stp -o
 1stp.out
```

Docking result potential energy file:

1stp.LigDockFin.ePL.res

#

```
NN    LigDockFinXXX.XXX.pdb         ePLtot   eVDW      eCoul     eHb        eSolv    eGeo
tempTAv
   1 1stp.LigDockFin001.001.    -56.1    -35.0   -12.9   -25.5    -8.0    25.4    49.4
- best eP_mode/rmsd_mode
   2 1stp.LigDockFin001.002.    -55.7   -34.5   -13.4   -25.3    -7.7    25.2    49.4
   3 1stp.LigDockFin001.003.    -55.7   -33.8   -13.9   -25.6    -7.8    25.4    49.4
   4 1stp.LigDockFin002.001.    -19.5   -21.3    -5.0    -8.6   -10.3    25.7    50.8
   5 1stp.LigDockFin002.002.    -18.5   -26.3    -4.6    -8.6    -7.5    28.5    50.8
   6 1stp.LigDockFin002.003.    -17.5   -20.7    -6.5    -9.5   -11.0    30.3    50.8
   7 1stp.LigDockFin003.001.    -19.9   -16.5    -9.4   -14.4   -10.2    30.6    50.6
   8 1stp.LigDockFin003.002.    -19.7   -19.5   -10.2   -14.6    -5.8    30.3    50.6
   9 1stp.LigDockFin003.003.    -19.3   -19.1    -8.5   -13.8    -6.3    28.5    50.6
  10 1stp.LigDockFin004.001.    -11.7   -12.3    -6.4    -9.5    -9.3    25.8    51.0
  11 1stp.LigDockFin004.002.    -11.6   -14.5    -4.3    -9.8   -10.2    27.2    51.0
  12 1stp.LigDockFin004.003.    -10.3   -12.8    -5.5    -9.6    -8.9    26.5    51.0
  13 1stp.LigDockFin005.001.    -12.9   -14.9    -7.7    -9.8    -6.9    26.3    51.1
  14 1stp.LigDockFin005.002.    -12.8   -16.8    -5.1    -4.9   -11.0    25.2    51.1
  15 1stp.LigDockFin005.003.    -11.0   -13.1    -9.3    -8.7    -7.2    27.3    51.1
  16 1stp.LigDockFin006.001.    -26.2   -17.1   -15.3   -18.3    -2.5    27.1    49.4
  17 1stp.LigDockFin006.002.    -26.2   -17.2   -15.4   -18.4    -2.6    27.3    49.4
  18 1stp.LigDockFin006.003.    -23.0   -17.9   -13.3   -14.9    -3.9    26.9    49.4
  19 1stp.LigDockFin007.001.    -30.4   -16.2   -17.2   -27.9     0.2    30.8    47.0
  20 1stp.LigDockFin007.002.    -29.2   -12.6   -18.3   -28.0     1.3    28.4    47.0
  21 1stp.LigDockFin007.003.    -28.1   -15.0   -18.5   -22.6    -0.2    28.2    47.0
```

..

NOTE!: three mode 1stp.LigDockFin001.001.pdb
                1stp.LigDockFin001.002.pdb
                1stp.LigDockFin001.003.pdb
are found by MD SA optimization from different initial orientations of the ligand

```
****************************************************************************
***********
#
#DOCK TEST#5
./3tpi : complex Trypsinogen/ILE-VAL dipeptide
#
#Run the mDynDock program:
by script
> . runMdynDock011.3tpi.sh
or by the command line
> $MDYNDOCKHOME/mDynDock011 -c 3tpi.IV.Prot.pdb -cL 3tpi.IV.notNative.pdb -i
MdynPar_d21.inp -sa SAprotocol_long.inp -mn 3tpi -o 3tpi.out
#
Docking result potential energy file:
3tpi.LigDockFin.ePL.res:
#
```

```
NN    LigDockFinXXX.XXX.pdb        ePLtot   eVDW      eCoul     eHb       eSolv     eGeo
tempTAv
   1 3tpi.LigDockFin001.001.    -80.1   -36.5   -24.1   -22.3   -10.1    12.8    51.7
   2 3tpi.LigDockFin001.002.    -79.8   -35.0   -23.8   -21.4   -10.6    11.0    51.7
   3 3tpi.LigDockFin001.003.    -77.7   -34.5   -24.2   -22.5   -10.3    13.9    51.7
   4 3tpi.LigDockFin002.001.    -47.9   -20.2   -13.8   -17.7    -4.8     8.5    51.9
   5 3tpi.LigDockFin002.002.    -45.9   -20.4   -17.1   -16.9    -4.0    12.5    51.9
   6 3tpi.LigDockFin002.003.    -44.1   -15.2   -19.0   -17.4    -3.4    11.0    51.9
   7 3tpi.LigDockFin003.001.    -54.0   -16.8   -21.2   -21.0    -6.5    11.5    49.1
   8 3tpi.LigDockFin003.002.    -53.9   -17.3   -21.0   -20.9    -6.4    11.7    49.1
   9 3tpi.LigDockFin003.003.    -52.8   -17.7   -20.5   -18.7    -5.6     9.7    49.1
  10 3tpi.LigDockFin004.001.    -81.8   -35.4   -24.5   -22.1   -10.5    10.7    50.4
best eP_mode/rmsd_mode
  11 3tpi.LigDockFin004.002.    -80.9   -36.4   -24.2   -22.4   -10.5    12.7    50.4
best eP_mode/rmsd_mode
  12 3tpi.LigDockFin004.003.    -72.0   -39.0   -17.8   -16.9   -10.8    12.5    50.4
  13 3tpi.LigDockFin005.001.    -40.9   -16.3   -16.4   -19.9   -10.9    22.7    49.4
  14 3tpi.LigDockFin005.002.    -39.6   -13.3   -16.1   -17.3    -7.8    14.9    49.4
  15 3tpi.LigDockFin005.003.    -38.3   -15.3   -12.8   -19.4    -1.8    11.0    49.4
  16 3tpi.LigDockFin006.001.    -43.1   -19.4   -11.1   -15.8    -9.1    12.3    52.0
  17 3tpi.LigDockFin006.002.    -41.4   -14.3    -9.6   -19.1   -12.4    13.9    52.0
  18 3tpi.LigDockFin006.003.    -37.9   -13.1   -14.6    -8.6    -9.4     7.9    52.0
  19 3tpi.LigDockFin007.001.    -41.1   -14.6   -22.9   -18.5    -3.3    18.2    49.0
  20 3tpi.LigDockFin007.002.    -40.7   -11.4   -23.3   -17.9    -3.4    15.3    49.0
  21 3tpi.LigDockFin007.003.    -40.0   -12.5   -23.7   -17.7    -2.5    16.5    49.0
...
```

```
#
******************************************************************************
***********
#DOCK TEST#6
./1hvr - complex HIV-1 protease/XK263 ligand : 1hvr PdB code
#
#Run the mDynDock program:
> . runMdynDock011.1hvr.sh

> $MDYNDOCKHOME/mDynDock011 -c 1hvr.P.pdb -cL 1hvr.Lig.pdb -i MdynPar_d21.inp
-sa SAprotocol_long.inp -mn 1hvr -o 1hvr.out
#
```

Docking result potential energy file:
1hvr.ePL.LigDockFin.res:

```
NN    LigDockFinXXX.XXX.pdb         ePLtot   eVDW      eCoul     eHb       eSolv     eGeo
tempTAv
   1 1hvr.LigDockFin001.001.     -54.0   -74.9   -13.1   -13.8      9.3    38.5    49.3
best eP_mode/rmsd mode
   2 1hvr.LigDockFin001.002.     -53.8   -76.0   -14.2   -13.9      9.2    41.1    49.3
best eP_mode/rmsd mode
   3 1hvr.LigDockFin001.003.    -52.9   -76.6   -12.8   -14.3     9.3    41.5    49.3
   4 1hvr.LigDockFin002.001.    -16.9   -56.4    -9.8   -11.7    10.0    51.0    48.2
   5 1hvr.LigDockFin002.002.    -10.4   -37.7    -9.2   -13.7    12.2    37.9    48.2
   6 1hvr.LigDockFin002.003.    -10.1   -44.5    -9.8   -13.0    12.2    45.1    48.2
   7 1hvr.LigDockFin003.001.    -24.1   -56.5   -11.2    -9.9     9.2    44.3    49.8
   8 1hvr.LigDockFin003.002.    -17.9   -57.7    -7.3    -8.7     9.1    46.8    49.8
   9 1hvr.LigDockFin003.003.    -12.2   -40.8    -9.5   -17.4    12.4    43.1    49.8
  10 1hvr.LigDockFin004.001.     -3.6   -35.4    -8.1    -9.1    14.6    34.4    49.5
  11 1hvr.LigDockFin004.002.     -2.7   -33.5    -4.0   -16.5    15.7    35.6    49.5
  12 1hvr.LigDockFin004.003.     -2.4   -36.0    -9.8   -11.2    14.6    40.0    49.5
  13 1hvr.LigDockFin005.001.     -2.5   -29.9    -7.8    -9.9    15.4    29.7    50.4
  14 1hvr.LigDockFin005.002.     -0.8   -38.8    -3.2    -9.1    14.3    35.9    50.4
  15 1hvr.LigDockFin005.003.      0.4   -37.4    -4.4    -7.7    14.6    35.4    50.4
  16 1hvr.LigDockFin006.001.     -6.6   -42.2    -6.4   -10.0    14.0    38.0    49.1
  17 1hvr.LigDockFin006.002.     -5.8   -35.5    -6.9   -13.0    14.1    35.5    49.1
  18 1hvr.LigDockFin006.003.     -4.1   -38.7    -5.9    -9.8    14.4    35.9    49.1
  19 1hvr.LigDockFin007.001.     -3.5   -35.0    -6.1   -13.2    14.0    36.8    50.7
  20 1hvr.LigDockFin007.002.     -3.2   -35.4    -6.5   -13.2    14.5    37.4    50.7
  21 1hvr.LigDockFin007.003.      0.9   -30.9    -5.0   -11.0    14.5    33.3    50.7
...
```

#

****
****

#

#DOCK TEST#7

./4phv

Complex HIV-1 protease with inhibitor VAC : PDB code 4phv

#

#Run the mDynDock program:

```
> . runMdynDock011.4phv.sh
> $MDYNDOCKHOME/mDynDock011 -c 4phv.P.pdb -cL 4phv.Lig.pdb -i MdynPar_d23.inp
-sa SAprotocol_long.inp -mn 4phv -o 4phv.out
```

#

Docking result potential energy file:
4phv.ePL.LigDockFin.res:

#

```
N   LigDockFinXXX.XXX.pdb      ePLtot   eVDW    eCoul   eHb     eSolv   eGeo    tempTAv
   1 4phv.LigDockFin001.001.    -57.1   -79.5   -12.2   -27.1     4.6    57.0    49.1
best eP_mode/rmsd_mode
   2 4phv.LigDockFin001.002.    -51.8   -81.4   -16.3   -24.6     4.6    65.9    49.1
   3 4phv.LigDockFin001.003.    -48.3   -68.8   -17.7   -32.5     6.6    64.1    49.1
   4 4phv.LigDockFin002.001.    -44.4   -74.5   -10.9   -20.7     6.2    55.4    49.9
   5 4phv.LigDockFin002.002.    -41.2   -62.0   -15.2   -32.4     6.7    61.8    49.9
   6 4phv.LigDockFin002.003.    -38.0   -60.6   -16.6   -27.3     6.8    59.7    49.9
   7 4phv.LigDockFin003.001.    -43.5   -69.4   -15.7   -28.6     6.9    63.2    48.9
   8 4phv.LigDockFin003.002.    -35.5   -67.0   -15.2   -25.2     7.5    64.4    48.9
   9 4phv.LigDockFin003.003.    -32.3   -59.6   -15.1   -23.6     7.5    58.5    48.9
  10 4phv.LigDockFin004.001.    -10.4   -31.4   -16.6   -28.5    14.8    51.3    49.9
  11 4phv.LigDockFin004.002.     -6.8   -32.0   -14.9   -28.2    14.0    54.4    49.9
  12 4phv.LigDockFin004.003.     -6.8   -41.0   -15.7   -19.4    11.0    58.3    49.9
  13 4phv.LigDockFin005.001.     -2.6   -44.5    -4.9   -19.2    11.8    54.1    50.1
  14 4phv.LigDockFin005.002.     -0.4   -39.1   -10.8   -18.1    13.0    54.5    50.1
  15 4phv.LigDockFin005.003.      1.0   -38.3    -9.9   -17.8    13.6    53.5    50.1
  16 4phv.LigDockFin006.001.    -20.0   -39.4   -20.1   -24.9    13.3    51.1    52.8
  17 4phv.LigDockFin006.002.    -18.5   -47.1   -12.5   -23.2    10.5    53.8    52.8
  18 4phv.LigDockFin006.003.    -16.0   -35.0   -19.1   -29.6    13.4    54.1    52.8
  19 4phv.LigDockFin007.001.    -28.8   -56.6   -13.0   -19.9     9.1    51.6    51.4
  20 4phv.LigDockFin007.002.    -27.6   -55.7   -15.1   -27.2     7.0    63.4    51.4
  21 4phv.LigDockFin007.003.    -24.2   -57.9   -10.1   -17.2     8.4    52.7    51.4
```

...

#

****

****

#DOCK TEST#8

./1hiv

Complex HIV-1 protease with inhibitor NOA Ligand (119 atoms): PDB code 1hiv

#

#run docking:

```
> . runMdynDock011.1hiv.sh
```

#

Docking with option $doLigDock=21 provides a small number (24) initial orientations for Lig and is unsufficient.

Docking with option $doLigDock=23 provides 72 initial Lig orientations and allows to find the same pose for two
different initial orientations.

#

Docking result potential energy file:

1hiv.ePL.DockFin.res:

```
NN    LigDockFinXXX.XXX.pdb        ePLtot   eVDW     eCoul    eHb       eSolv    eGeo
tempTAv
   1 1hiv.LigDockFin001.001.    -84.9    -86.6    -28.0    -41.4     10.4     60.7     51.3
best eP_mode/rmsd
   2 1hiv.LigDockFin001.002.    -84.6    -91.0    -28.1    -41.3     10.4     65.5     51.3
best eP_mode/rmsd
   3 1hiv.LigDockFin001.003.    -70.4    -78.5    -29.8    -39.8      7.7     70.0     51.3
   4 1hiv.LigDockFin002.001.    -78.3    -80.3    -29.0    -40.8     10.9     61.0     50.1
   5 1hiv.LigDockFin002.002.    -74.1    -82.0    -28.9    -44.6     10.2     71.3     50.1
   6 1hiv.LigDockFin002.003.    -54.7    -67.2    -26.8    -40.8     12.0     68.1     50.1
   7 1hiv.LigDockFin003.001.    -15.0    -38.9    -22.9    -24.6     18.4     52.8     50.9
   8 1hiv.LigDockFin003.002.     -8.9    -37.8    -22.6    -22.9     19.4     55.0     50.9
   9 1hiv.LigDockFin003.003.     -8.3    -39.9    -19.9    -20.0     19.0     52.5     50.9
  10 1hiv.LigDockFin004.001.    -24.1    -48.2    -23.7    -32.6     16.3     64.1     49.9
  11 1hiv.LigDockFin004.002.    -23.9    -47.8    -20.0    -35.7     22.0     57.7     49.9
  12 1hiv.LigDockFin004.003.    -19.2    -45.8    -22.1    -30.0     17.2     61.5     49.9
  13 1hiv.LigDockFin005.001.    -18.8    -50.8    -21.0    -24.2     18.1     59.1     50.8
  14 1hiv.LigDockFin005.002.    -15.7    -42.8    -25.0    -29.7     18.7     63.2     50.8
  15 1hiv.LigDockFin005.003.    -14.1    -39.7    -23.7    -29.6     18.3     60.6     50.8
  16 1hiv.LigDockFin006.001.    -38.1    -61.4    -24.1    -28.0     17.4     58.0     50.9
  17 1hiv.LigDockFin006.002.    -34.9   -62.4   -25.7   -29.8    16.6    66.4    50.9
  18 1hiv.LigDockFin006.003.    -33.5   -60.3   -23.5   -31.2    15.1    66.3    50.9
  19 1hiv.LigDockFin007.001.    -40.2   -67.9   -24.6   -31.8     9.8    74.4    50.0
  20 1hiv.LigDockFin007.002.    -38.3   -75.4   -22.5   -20.7     6.6    73.6    50.0
  21 1hiv.LigDockFin007.003.    -36.2   -70.3   -20.6   -33.4     8.0    80.1    50.0
  22 1hiv.LigDockFin008.001.    -16.4   -45.3   -21.6   -22.2    14.7    57.9    51.6
 ....
  34 1hiv.LigDockFin012.001.     -84.4   -88.7   -29.7   -40.6    10.4    64.1    51.2
best eP_mode/rmsd
  35 1hiv.LigDockFin012.002.    -80.4   -84.1   -31.8   -42.1     8.5    69.0    51.2
  36 1hiv.LigDockFin012.003.    -69.9   -83.3   -31.2   -39.7     9.4    74.9    51.2
```

****
****

RESUME :

1) all shown test examples of docking find a set of docking modes, i.e. files
   LigDockFinXXX.XXX.pdb
2) the mode with minimal potential energy of Protein-Lig interactions in the set of files
   LigDockFinXXX.XXX.pdb is the mode close to the respective native ligand structure in the complex.
   The RMSD of the best docking mode from the native are within 1 - 2 A.
3) The current docking method does not guarantee a finding of the best docking solution in the one RUN. The best docking solution can be obtained by docking with different set of initial Lig orientations by oprion $doLigDock=21(set24), 22(set144) or 23(set72 orientations)
 and simulated annealing protocol -sa saProtocol
#

TYPICAL DOCKING Protocol

example for the complex 1etr
see: ~/MDynDock011/dockProtocol/1etr

1) make directories

./P	isolated protein
./PL	ProtLigand complex
./iLg	isolated Lig

copy initial PDBfile 1ETR.pdb from the PDB data base to the ./1etr

2. create initial protein structure files

remove all water molecules, ions and Ligand atoms from 1ETR.pdb
get the initial protein structure file 1etr.P.0.pdb

3. get the initial Ligand structure file = 1etr.L.0.pdb, i.e.
copy from PDB or build by Molcular Constructor

4. create full atom protein structure,
Protein structure may have
missing side chain heavy atoms for some residues,
missing hydrogens

copy file 1etr.P.0.pdb ./P

RUN mDynQ011 to add all missing side chain heavy atoms
and to add all hydrogens

```
> $MDYN011HOME/mDynQ011 -c 1etr.P.0.pdb -i s1_MdynPar.inp -o 1etr.P.0.out
```

Program will create all atom energy minimized protein structere PDB file
molEnOpt0001.pdb
copy this file to 1etr.P.eOp.pdb - it is all atom protein structure file

5. Create all atom Ligand structure and Topology file for the Ligand

```
> cd ./iLg

#  make new directory
> mkdir mTop

# copy ligand heavy atom structure to ./iLg/mTop  directory
>cp  1etr.L.0.pdb ./iLg/mTop
```

# make ligand heavy atom file with added Htag, i.e. number of H-atoms to be added to heavy atoms

#

```
 > cp 1etr.L.0.pdb 1etr.L.Htag.pdb
```

file 1etr.L.Htag.pdb **should be edited**, the number of hydrogen atoms attached to heavy atoms should be added at the end of ATOM line, see example file: 1etr.L.Htag.pdb

RECOMENDED: visualization of Lig structure to find out hydrogens to be added to each heavy atom

# make the ligand topology file and all atom (with added H) ligand structure

6. Run the program mTopoHQ

```
    > ~/mTopo011/mTopoHQ -c 1etr.L.Htag.pdb -i mTopoInProtocol.inp -h 1etr.L.h.0.pdb -mt 1etr.MQI.mTop.dat -o 1etr.MQI.mTop.out
```

inputDAta files: 1etr.L.Htag.pdb

mTopoInProtocol.inp - the command file (see mTopoReadMe.txt)

RESULT files:

1etr.L.h.0.pdb - all atom PDB file of Ligand structure with added Hydrogens

1etr.MQI.mTop.dat - ligandMolecule Topology file with calculated atomic charges

copy 1etr.L.h.0.pdb 1etr.MQI.mTop.dat ../PL directory

7. Run docking procedure for the ligand to the protein structure

# copy energy optimized protein structure to the ./PL directory

```
   cd  ./PL
   copy ./P/1etr.P.eOp.pdb  ./PL
```

#copy ligand structure and ligand topology files

```
   copy ./iLg/mTop/1etr.L.h.0.pdb  ./PL
   copy ./iLg/mTop/1etr.MQI.mTop.dat ./PL
```

Run Docking procedure

make dir ./d22

Prepare the command files for docking

MdynPar_1etr.inp
SAprotocol_long.inp

NOTE! see examples how to make the command files for docking

copy Ligand topology file ../iLg/mTop/1etr.MQI.mTop.dat ./PL

Run the docking :

```
>   $MDYNDOCKHOME/mDynDock011   -c   1etr.PL.eOp.pdb   -cL   1etr.L.h.0.pdb   -i
MdynPar_1etr.inp -tL 1etr.MQI.mTop.dat
                            -sa SAprotocol_long.inp  -o 1etr.PL.d22.out
```

command line parameters:

```
    -c 1etr.PL.eOp.pdb       -  file for Protein allAtomstructure
    -cL 1etr.L.h.0.pdb       -  Ligand allAtom structure
    -i MdynPar_1etr.inp      -  main command file
      -sa  SAprotocol_long.inp   -  simulated  annealing  protocol  for  ligand
binding mode optimization
    -tL ../1etr.MQI.mTop.dat - ligand molecule topology file
    -o 1etr.PL.d22.out       - output file
```

UNDERSTANDING DOCKING RESULT

The docking program calculates

1) a list of low-resolution ligand binding sites with score

2) a list of high resolution binding sites with protein-ligand potential energy of interaction

8.1. 1etr.LigBSiteOnSAS01.pdb - low-resolution ligand binding sites with score

```
#LigBindGridOnSAS:                 Position XYZ          ContactScore
ATOM      1  LBSt         1     -30.336 -33.330  32.557   16
ATOM      2  LBSt         2     -56.016 -22.175  44.997   15
ATOM      3  LBSt         3     -51.339 -36.585  28.474   15
ATOM      4  LBSt         4     -40.564 -24.946  19.907   14
ATOM      5  LBSt         5     -37.127 -30.225  41.166   13
ATOM      6  LBSt         6     -41.741 -43.508  38.795   12
ATOM      7  LBSt         7     -59.214 -22.328  20.087   12
ATOM      8  LBSt         8     -45.261  -5.030  33.232   10
ATOM      9  LBSt         9     -55.434 -15.925  45.268   10
ATOM     10  LBSt        10     -64.919 -20.800  33.039   10
ATOM     11  LBSt        11     -51.170  -4.853  38.822    9
ATOM     12  LBSt        12     -23.682 -28.532  19.464    9
ATOM     13  LBSt        13     -29.671 -28.992  38.703    9
ATOM     14  LBSt        14     -53.333 -36.116  50.956    8
ATOM     15  LBSt        15     -48.613 -37.657  40.769    8
ATOM     16  LBSt        16     -44.875  -7.795  20.803    8
ATOM     17  LBSt        17     -38.440 -14.427  54.697    8
ATOM     18  LBSt        18     -26.059 -31.899  26.808    8
ATOM     19  LBSt        19     -34.803 -28.383  46.800    7
......................................
ATOM     70  LBSt        70     -56.365 -35.639  25.176    4
ATOM     71  LBSt        71     -65.405 -30.342  25.632    4
ATOM     72  LBSt        72     -59.910 -27.593  13.815    4
ATOM     73  LBSt        73     -47.055 -18.638   8.540    4
##
```

8.2. 1etr.LigDockFin_ePL.res file are the energy of protein/ligand interactions for high-resolution docking (all atom Ligand)

#example:

```
  1 1etr.LigDockFin001.001.    -43.8   -42.6   -10.7   -29.2     7.8    30.8    51.1
  2 1etr.LigDockFin001.002.    -42.2   -45.0   -13.3   -23.7     3.6    36.1    51.1
  3 1etr.LigDockFin001.003.    -42.0   -42.5   -13.9   -30.0     7.2    37.4    51.1
  4 1etr.LigDockFin002.001.    -44.5   -33.2   -14.8   -31.1     2.7    32.0    50.9
  5 1etr.LigDockFin002.002.    -43.2   -37.5   -14.5   -30.7     2.5    37.1    50.9
  6 1etr.LigDockFin002.003.    -41.3   -34.2   -13.9   -32.1     2.8    36.2    50.9
  7 1etr.LigDockFin003.001.    -34.7   -39.4   -12.1   -19.5     0.2    36.1    51.3
  8 1etr.LigDockFin003.002.    -34.3   -35.1   -14.2   -28.4     6.2    37.2    51.3
  9 1etr.LigDockFin003.003.    -33.8   -37.6   -12.0   -26.2      6.2    35.8    51.3
 10 1etr.LigDockFin004.001.    128.8    48.8     0.4   -10.8     -1.3    91.7    49.7
 11 1etr.LigDockFin004.002.    196.3   103.3    -6.6    -7.9     -2.8   110.3    49.7
 12 1etr.LigDockFin004.003.    226.5   120.7    -2.1    -8.4      1.4   114.8    49.7
 13 1etr.LigDockFin005.001.     -56.4   -53.0   -16.9   -30.0      3.1    40.5    51.6
#best ePot ~ nativeMode
 14 1etr.LigDockFin005.002.     -54.9   -54.3   -15.5   -27.3      3.3    38.9    51.6
& best RMSD from Native
 15 1etr.LigDockFin005.003.    -52.3   -49.0   -15.7   -33.1      4.4    41.1    51.6
 16 1etr.LigDockFin006.001.    -32.5   -34.4   -10.1   -31.1      3.9    39.2    48.5
 17 1etr.LigDockFin006.002.    -30.3   -36.9   -11.4   -23.1      5.1    36.0    48.5
 18 1etr.LigDockFin006.003.    -29.7   -37.3   -11.3   -22.0      4.7    36.3    48.5
 19 1etr.LigDockFin007.001.    -30.8   -39.5    -5.1   -20.8      1.2    33.4    47.1
 20 1etr.LigDockFin007.002.    -30.2   -39.9    -5.2   -19.4      1.4    32.9    47.1
 21 1etr.LigDockFin007.003.    -30.1   -39.9    -7.8   -20.1      1.4    36.3    47.1
 22 1etr.LigDockFin008.001.    -32.8   -40.5    -9.7   -26.0      5.2    38.1    49.6
 23 1etr.LigDockFin008.002.    -30.1   -36.2   -11.7   -29.5      3.6    43.8    49.6
 24 1etr.LigDockFin008.003.    -29.7   -38.5   -10.0   -20.2      5.2    33.8    49.6
 25 1etr.LigDockFin009.001.    -40.9   -35.3   -15.9   -29.8      2.3    37.7    47.4
 26 1etr.LigDockFin009.002.    -38.0   -36.9   -12.1   -31.8      2.8    40.0    47.4
 27 1etr.LigDockFin009.003.    -36.6   -37.7    -8.0   -27.2      5.9    30.3    47.4
 28 1etr.LigDockFin010.001.    -26.0   -34.9    -9.7   -21.4      4.3    35.8    48.8
 29 1etr.LigDockFin010.002.    -22.1   -33.4   -11.8   -22.1      4.7    40.5    48.8
 30 1etr.LigDockFin010.003.    -21.0   -30.1    -8.8   -19.3      4.7    32.5    48.8
 31 1etr.LigDockFin011.001.    -35.9   -39.3   -10.9   -26.4      4.8    36.0    51.1
 32 1etr.LigDockFin011.002.    -34.8   -33.5   -13.6   -25.5      6.1    31.7    51.1
 33 1etr.LigDockFin011.003.    -34.4   -33.4   -13.1   -25.8      5.7    32.3    51.1
 34 1etr.LigDockFin012.001.    -33.1   -34.0   -10.8   -27.0      5.3    33.4    49.6
 35 1etr.LigDockFin012.002.    -30.4   -31.7   -10.9   -26.6      6.1    32.7    49.6
 36 1etr.LigDockFin012.003.    -30.3   -39.0    -8.3   -22.9      4.6    35.3    49.6
 37 1etr.LigDockFin013.001.    -47.1   -43.3   -14.9   -35.6      3.7    43.1    48.9
 38 1etr.LigDockFin013.002.    -45.4   -38.8   -14.7   -32.2      4.3    35.9    48.9
 39 1etr.LigDockFin013.003.    -44.3   -37.8   -14.3   -32.7      4.6    35.9    48.9
 40 1etr.LigDockFin014.001.    -22.5   -27.8    -9.4   -25.2      5.8    34.2    49.3
 41 1etr.LigDockFin014.002.    -22.3   -26.7   -10.3   -28.2      7.4    35.4    49.3
 42 1etr.LigDockFin014.003.    -22.3   -28.4   -10.2   -25.5      5.9    36.0    49.3
 43 1etr.LigDockFin015.001.    122.6    46.9    -4.5   -15.8      3.0    93.0    50.9
 44 1etr.LigDockFin015.002.    128.1    45.5    -2.3    -9.2      3.2    90.9    50.9
 45 1etr.LigDockFin015.003.    137.8    49.3    -7.8   -15.7      1.8   110.2    50.9
```

#

8.3 A list of high-resolution ligand structures

```
LigDockFin001.001.pdb
LigDockFin001.002.pdb
LigDockFin001.003.pdb
LigDockFin005.002.pdb
LigDockFin005.003.pdb
LigDockFin006.001.pdb
```

```
   LigDockFin002.001.pdb          LigDockFin006.002.pdb
   LigDockFin002.002.pdb          LigDockFin006.003.pdb
   LigDockFin002.003.pdb
   .... etc.
```

Where, the first number 005. is the number of low resolution binding site taken for high-resolution docking

the second number .001 .002 .003 are the best three orientations found by global optimization

by the MD simulated annealing.

The 1etr.LigDockFin005.001.pdb has the minimal energy of Ligand interaction with protein this mode should taken as the best #1 in the blind docking procedure.

NOTE, that RMSD of the first two low-energy structures 1etr.LigDockFin005.001.pdb 1etr.LigDockFin005.002.pdb

from the native ligand structure 1etr.L.h.0.pdb in the protein-ligand complex are about ~ 1.0 A.

#END

Calculation of the molecular topology file for a new Ligand

# mTopo program

program mTopo calculates molecular topology file newLigand_mTopo.dat

for a new Ligand molecule from the Ligand PDB file newLigXYZ.pdb

The calculated topology file newLigand_mTopo.dat can be added to the Library of Topology files of the program mDynDock011 to perform DOCKING calculation of this Ligand with proteins.

RUN the program mTopo

Program mTopo is executed by command line:

#> mTopo -i inProtcol -c inPDB -mt molecTopoFile -mn molName -h -o runOutFile

INPUT files:

```
-i inProtcol : input protocol file defines protocol of calculation,
            default name inProtcol = ./mTopoInPar.inp
              if key -i is absent, the program use the default file
./mTopoInPar.inp
```

```
          in the current (job) directory
-c inPDB   : the PDB file of the newLigand. The PDB file  can have MISSING hydrogens.
            default name inPDB = ./molecIn.pdb
```

RESULT files:

```
-mt molecTopoFile : calculated molecular topology file
             default name = ./molecTopo.dat

-o runOutFile     : intermediate calculation result output file
                   default name = ./mTopo.out
```

EXAMPLE of INPUT files

```
inProtcol:
---------------------
#mTopoInProtocol.inp
$LigName XK2               ! LIG name in result molecTopoFile
$ADDH                      ! flag to add H atoms
$AtRename                  ! will rename atoms compare to its name in
                              the inputPDB
$QTOT   0.0                !total charge Q of the molecule, i.e.
                            = 0.0, 1.0, -1.0, etc.
END
---------------------
```

inPDB:

NOTE: if the file inProtcol has the keyword $ADDH,
then the inPDB file should have USER defined number of added Hydrogens in the ATOM line
in the positions 56-60, just after XYZ coordinates.
Thise number of Hydrigens for ATOMs will be added by the program mTopo

```
-------------------------------------------------------
REMARK: Ligand PDB:
#23456789012345678901234567890123456789012345678901234567890
#aaaaaiiiiixxaaaaAAAAaiiiixxxxffffffffFFFFFFFFffffffffxnhh
ATOM   3119  O1  XK2 A 199      -7.238  14.948  28.050
ATOM   3120  C1  XK2 A 199      -8.308  15.545  27.895
ATOM   3121  N2  XK2 A 199      -8.395  16.130  26.643
ATOM   3122  C2  XK2 A 199      -8.222  15.124  25.546 2
ATOM   3123  C20 XK2 A 199      -8.328  15.687  24.139
ATOM   3124  H21 XK2 A 199      -7.293  14.564  25.591
ATOM   3125  H22 XK2 A 199      -9.012  14.392  25.644
ATOM   3126  C21 XK2 A 199      -9.603  15.774  23.537 1
ATOM   3127  HC21XK2 A 199     -10.484  15.486  24.092
ATOM   3128  C22 XK2 A 199      -9.691  16.151  22.192 1
ATOM   3129  C23 XK2 A 199      -8.532  16.543  21.512
ATOM   3130  HC22XK2 A 199     -10.656  16.202  21.690
ATOM   3131  C24 XK2 A 199      -8.635  16.965  20.203 1
ATOM   3132  C25 XK2 A 199      -7.487  17.309  19.501 1
ATOM   3133  H25 XK2 A 199      -7.603  17.609  18.458
ATOM   3134  C26 XK2 A 199      -6.233  17.295  20.127 1
ATOM   3135  H26 XK2 A 199      -5.335  17.519  19.578
ATOM   3136  C27 XK2 A 199      -6.147  16.901  21.457 1
ATOM   3137  H27 XK2 A 199      -5.181  16.885  21.948
ATOM   3138  C28 XK2 A 199      -7.285  16.509  22.147
ATOM   3139  C29 XK2 A 199      -7.164  16.080  23.469 1
ATOM   3140  H29 XK2 A 199      -6.213  16.015  23.956
ATOM   3141  C3  XK2 A 199      -8.623  17.592  26.549
ATOM   3142  C31 XK2 A 199      -7.483  18.393  27.185 2
ATOM   3143  H331XK2 A 199      -7.631  19.470  27.036
ATOM   3144  H332XK2 A 199      -7.484  18.146  28.244
ATOM   3145  C32 XK2 A 199      -6.139  18.105  26.566
ATOM   3146  C33 XK2 A 199      -5.159  17.416  27.284 1
ATOM   3147  H33 XK2 A 199      -5.334  17.036  28.274
ATOM   3148  C34 XK2 A 199      -3.893  17.231  26.751 1
ATOM   3149  H34 XK2 A 199      -3.149  16.745  27.351
ATOM   3150  C35 XK2 A 199      -3.626  17.620  25.445 1
ATOM   3151  H35 XK2 A 199      -2.655  17.435  25.051
ATOM   3152  C36 XK2 A 199      -4.621  18.240  24.694 1
ATOM   3153  H36 XK2 A 199      -4.431  18.485  23.661
ATOM   3154  C37 XK2 A 199      -5.857  18.546  25.264 1
ATOM   3155  H37 XK2 A 199      -6.566  19.106  24.694
ATOM   3156  H3  XK2 A 199      -8.657  17.893  25.492
ATOM   3157  C4  XK2 A 199     -10.020  17.960  27.017 1
ATOM   3158  C5  XK2 A 199     -10.338  17.646  28.475
ATOM   3159  O4  XK2 A 199     -10.106  19.343  26.902 1
ATOM   3160  HO4 XK2 A 199     -11.013  19.637  26.930
ATOM   3161  O5  XK2 A 199     -11.549  18.269  28.808 1
ATOM   3162  HO5 XK2 A 199     -11.386  19.223  28.899
ATOM   3163  C6  XK2 A 199     -10.531  16.169  28.829 1
ATOM   3164  H6  XK2 A 199     -11.101  16.230  29.748
ATOM   3165  N7  XK2 A 199      -9.227  15.515  28.880
ATOM   3166  C61 XK2 A 199     -11.468  15.454  27.835 2
ATOM   3167  H611XK2 A 199     -10.934  15.195  26.915
ATOM   3168  H612XK2 A 199     -12.270  16.125  27.588
ATOM   3169  C62 XK2 A 199     -12.135  14.199  28.390
ATOM   3170  C63 XK2 A 199     -11.527  12.951  28.235 1
ATOM   3171  H63 XK2 A 199     -10.597  12.868  27.683
ATOM   3172  C64 XK2 A 199     -12.110  11.806  28.805 1
ATOM   3173  H64 XK2 A 199     -11.569  10.872  28.708
ATOM   3174  C65 XK2 A 199     -13.331  11.912  29.477 1
ATOM   3175  H65 XK2 A 199     -13.744  11.021  29.933
ATOM   3176  C66 XK2 A 199     -13.975  13.152  29.576 1
ATOM   3177  H66 XK2 A 199     -14.935  13.211  30.077
ATOM   3178  C67 XK2 A 199     -13.381  14.282  29.035 1
ATOM   3179  H67 XK2 A 199     -13.909  15.244  29.110
ATOM   3180  C7  XK2 A 199      -8.923  14.733  30.058 2
ATOM   3181  HC71XK2 A 199      -7.885  14.809  30.412
ATOM   3182  HC72XK2 A 199      -9.021  13.651  29.898
ATOM   3183  C70 XK2 A 199      -9.644  14.957  31.348
ATOM   3184  C71 XK2 A 199      -9.056  15.739  32.338 1
ATOM   3185  H71 XK2 A 199      -8.123  16.262  32.160
ATOM   3186  C72 XK2 A 199      -9.666  15.780  33.599
ATOM   3187  C73 XK2 A 199      -9.033  16.471  34.629 1
ATOM   3188  H73 XK2 A 199      -8.125  17.012  34.440
ATOM   3189  C74 XK2 A 199      -9.550  16.421  35.925 1
ATOM   3190  H74 XK2 A 199      -9.036  16.949  36.704
ATOM   3191  C75 XK2 A 199     -10.768  15.770  36.182 1
ATOM   3192  H75 XK2 A 199     -11.222  15.725  37.157
ATOM   3193  C76 XK2 A 199     -11.430  15.116  35.148 1
ATOM   3194  H76 XK2 A 199     -12.346  14.545  35.374
ATOM   3195  C77 XK2 A 199     -10.855  15.085  33.871
ATOM   3196  C78 XK2 A 199     -11.449  14.346  32.857 1
ATOM   3197  H78 XK2 A 199     -12.386  13.805  33.019
ATOM   3198  C79 XK2 A 199     -10.830  14.258  31.613 1
ATOM   3199  H79 XK2 A 199     -11.285  13.630  30.878
ATOM   3200  H24 XK2 A 199      -9.593  16.999  19.740
ATOM   3201  HC4 XK2 A 199     -10.732  17.510  26.342
ATOM   3202  HC5 XK2 A 199      -9.573  18.037  29.145
END
------------------------------


RESULT file: mTopoFile.dat


--------------------------------
```

```
# LIG TOPO generated by mTopo
$MTRES
#    Res  Typ   NAT    Q
XK2  ISO  LG    85    0.00  1
#------------------------
#  L  <--------------- J<--K<--I      bndLJ    angLJK  tangLJKI   qAtom
#
   1  DUMM  DU    M    0  -1  -2      0.0000     0.00     0.00   0.0000
   2  DUMM  DU    M    1   0  -1      1.5000     0.00     0.00   0.0000
   3  DUMM  DU    M    2   1   0      1.5000   150.00     0.00   0.0000
   4  O1    O     M    3   2   1      1.5000   150.00     0.00  -0.6002
   5  C1    C     M    4   3   2      1.2350   107.82   111.00   0.6925
   6  N2    N     M    5   4   3      1.3847   111.85   -39.26  -0.4302
   7  N7    N     M    5   4   3      1.3475   119.23   138.35  -0.4472
   8  C2    CT    M    6   5   4      1.4985   111.78    56.90  -0.1622
   9  C3    CK    M    6   5   4      1.4827   118.92  -122.61   0.2001
  10  C6    CT    M    7   5   4      1.4597   125.01  -179.74   0.0575
  11  C7    CT    M    7   5   4      1.4462   117.65    -2.28  -0.1564
  12  HC21  HC    E    8   6   5      1.0000   108.09    58.24   0.1188
  13  HC22  HC    E    8   6   5      1.0000   108.09   -60.19   0.1891
  14  C20   CB    M    8   6   5      1.5192   114.91   179.03   0.0837
  15  C31   CT    M    9   6   5      1.5316   112.02    61.87  -0.4258
  16  C4    CT    M    9   6   5      1.5186   111.15   -68.97   0.1486
  17  C5    C     M   16   9   6      1.5250   115.89    62.37   0.2109
  18  HC61  HC    E   10   7   5      1.0000   109.60  -171.79   0.1035
  19  C61   CT    M   10   7   5      1.5418   110.96    70.79  -0.3975
  20  HC71  HC    E   11   7   5      1.0000   106.41    39.32   0.1809
  21  HC72  HC    E   11   7   5      1.0000   106.41   -77.39   0.1556
  22  C70   CB    M   11   7   5      1.4947   121.37   160.96   0.0638
  23  C21   CA    M   14   8   6      1.4127   118.72   -87.41  -0.1336
  24  C29   CA    M   14   8   6      1.3994   119.31    93.76  -0.1990
  25  H311  HC    E   15   9   6      1.0000   108.54   -62.15   0.2007
  26  H312  HC    E   15   9   6      1.0000   108.54   178.92   0.2423
  27  C32   CB    M   15   9   6      1.5075   113.15    58.38   0.1409
  28  HC41  HC    E   16   9   6      1.0000   106.58   -56.12   0.0625
  29  O4    OH    M   16   9   6      1.3904   105.81   179.14  -0.7267
  30  O5    OH    M   17  16   9      1.4020   108.40   168.90  -0.7492
  31  H611  HC    E   19  10   7      1.0000   108.23   -40.06   0.1873
  32  H612  HC    E   19  10   7      1.0000   108.23  -158.65   0.1979
  33  C62   CB    M   19  10   7      1.5258   114.37    80.64   0.1042
  34  C71   CA    M   22  11   7      1.3919   119.62   -98.29  -0.1940
  35  C79   CA    M   22  11   7      1.4019   119.77    89.02  -0.1544
  36  H211  HA    E   23  14   8      1.0000   120.56     7.02   0.1063
```

```
37  C22    CA    M   23  14   8      1.3996   118.87  -172.98  -0.1272
38  C23    C     M   37  23  14      1.3998   119.35    -5.50   0.0463
39  H291   HA    E   24  14   8      1.0000   121.05    -4.19   0.1524
40  C33    CA    M   27  15   9      1.3967   120.58  -110.54  -0.1569
41  C37    CA    M   27  15   9      1.4033   120.01    70.33  -0.1656
42  HO41   HO    E   29  16   9      1.0000   109.47   179.98   0.4878
43  HO51   HO    E   30  17  16      1.0000   109.47   180.00   0.4002
44  C63    CA    M   33  19  10      1.3969   120.28   -90.97  -0.1608
45  C67    CA    M   33  19  10      1.4055   120.39    90.49  -0.1621
46  H711   HA    E   34  22  11      1.0000   120.90     8.67   0.1272
47  C72    CB    M   34  22  11      1.4014   118.20  -171.33   0.0643
48  C77    CA    M   47  34  22      1.4038   121.91    -1.48   0.0416
49  H791   HA    E   35  22  11      1.0000   119.55    -6.47   0.1428
50  H221   HA    E   37  23  14      1.0000   120.32   174.50   0.1077
51  C28    C     M   38  37  23      1.3998   120.69     2.61   0.0667
52  C27    CA    M   51  38  37      1.3874   119.88   179.98  -0.1432
53  H331   HA    E   40  27  15      1.0000   119.70     4.45   0.1556
54  C34    CA    M   40  27  15      1.3860   120.60  -175.55  -0.0657
55  C35    CA    M   54  40  27      1.3886   119.99    -6.08  -0.1096
56  H371   HA    E   41  27  15      1.0000   120.39     1.68   0.1416
57  H631   HA    E   44  33  19      1.0000   119.93    -3.18   0.1219
58  C64    CA    M   44  33  19      1.4056   120.15   176.82  -0.0823
59  C65    CA    M   58  44  33      1.3977   119.70     2.82  -0.1088
60  H671   HA    E   45  33  19      1.0000   119.64     1.88   0.1164
61  C73    CA    M   47  34  22      1.3925   118.83   174.94  -0.1429
62  C78    C     M   48  47  34      1.3882   118.98    -0.60  -0.1256
63  H781   H     E   62  48  47      1.0000   120.13  -177.18   0.1069
64  C24    CA    M   38  37  23      1.3792   119.01  -177.39  -0.1377
65  C25    CA    M   64  38  37      1.3889   119.59  -177.60  -0.0848
66  H271   HA    E   52  51  38      1.0000   119.82   178.34   0.1078
67  H341   HA    E   54  40  27      1.0000   120.00   173.92   0.0959
68  C36    CA    M   55  54  40      1.3923   119.65     2.36  -0.0744
69  H361   HA    E   68  55  54      1.0000   119.65  -176.18   0.0868
70  H641   HA    E   58  44  33      1.0000   120.15  -177.18   0.0924
71  C66    CA    M   59  58  44      1.4008   120.18     0.35  -0.0780
72  H661   HA    E   71  59  58      1.0000   120.10   178.34   0.0889
73  H731   HA    E   61  47  34      1.0000   119.98     6.18   0.1107
74  C74    CA    M   61  47  34      1.3962   120.03  -173.82  -0.0843
75  C75    CA    M   74  61  47      1.4048   120.50    -5.33  -0.0873
76  H241   HA    E   64  38  37      1.0000   120.21     2.40   0.1097
77  C26    CA    M   65  64  38      1.4016   120.75    -3.15  -0.0863
78  H261   HA    E   77  65  64      1.0000   120.47  -178.52   0.0930
79  H351   HA    E   55  54  40      1.0000   120.18  -177.64   0.0977
```

```
  80  H651  HA    E   59  58  44       1.0000   119.91  -179.65   0.0971
  81  H741  HA    E   74  61  47       1.0000   119.75   174.67   0.0940
  82  C76   CA    M   75  74  61       1.3911   119.64     3.10  -0.1363
  83  H761  HA    E   82  75  74       1.0000   120.24  -178.31   0.1072
  84  H251  HA    E   65  64  38       1.0000   119.63   176.85   0.0942
  85  H751  HA    E   75  74  61       1.0000   120.18  -176.90   0.0941
##QTOT =    0.000000
#xxxiiii--
LOOP    7
#     nLoop   nTYPEbond
D:  77  52   1
#S C26 C27
D:  51  24   2
#S C28 C29
D:  68  41   3
#S C36 C37
D:  17  10   4
#S C5  C6
D:  71  45   5
#S C66 C67
D:  82  48   6
#S C76 C77
D:  62  35   7
#S C78 C79
##
DONE
#
```

MolMech

In the current version of the program, the PDB file with coordinates of atoms in a protein in the input data. The coordinates may be retrieved from the file or PDB database. For computation, indicate the chain identifier, given in the PDB file.

The program automatically prepares the file with topology of the molecule, containing AMBER force field parameters. The program uses this file in further calculations of molecular mechanical minimization. A standard AMBER and/or user topology database of individual residues is used for creating this topology file. AMBER parameters file is used for determining the constants of potential energy function, such as equilibrium bond lengths, angles, dihedral angles, their force constants, non-bonded 6-12 parameters, and H-bond 10-12 parameters.

Minimization stops after 50 iterations.

The output data are the coordinates of the atoms of protein chain after minimization in PDB format.

Output example:

```
HEADER    SoftBerry molecular mechanic Ver. 1.0
REMARK   1
REMARK   1 Charge modification is NOT performed.
REMARK   1 NO periodic boundaries are applied.
REMARK   1 Non-bonded interactions evaluated normally.
REMARK   1 Energy is reported in Kcal/mol
REMARK   1 Complete interaction is calculated.
REMARK   1 NB pairlist generated in residue-residue basis.
REMARK   1 No pair list will be generated.
REMARK   1 NB list updated every 10 steps.
REMARK   1 Buffer region updates every 1 steps.
REMARK   1 Constant dielectric function used.
REMARK   1 Solvent pointer = 142.
REMARK   1 No water model chosen.
REMARK   1 NB cutoff distance =     8.0000 Angstroms.
REMARK   1 1,4 non-bonds divided by     2.0000.
REMARK   1 1,4 electrostatics divided by     2.0000.
REMARK   1 The dielectric constant =     1.0000.
REMARK   1 The buffer cutoff is    8.00000 Angstroms.
REMARK   1 CAP Option is inactivated.
REMARK   1
REMARK   1 The number of degrees of freedom = 6426.
REMARK   1 INITIAL CONDITIONS OF SYSTEM:
REMARK   1
REMARK   1 Potential Energy = -4643.602515
REMARK   1 Non-bond         = -784.604532
REMARK   1 H-bond           = 0.000000
REMARK   1 Electrostatic    = -10490.096084
REMARK   1 Bond             = 183.712294
REMARK   1 Angle            = 715.484007
REMARK   1 Dihedral         = 557.877658
REMARK   1 1,4 Non-bonded   = 721.197306
REMARK   1 1,4 Electrostatic= 4452.826836
REMARK   1
REMARK   1 MINIMIZATION TERMINATED  : Exceeded maximum number of cycles
REMARK   1 Number of function calls 102
REMARK   1 Number of iterations 50
REMARK   1
REMARK   1 Potential Energy = -6031.148428
REMARK   1 Non-bond         = -1078.280106
REMARK   1 H-bond           = 0.000000
REMARK   1 Electrostatic    = -10870.756945
REMARK   1 Bond             = 38.980831
REMARK   1 Angle            = 364.506930
REMARK   1 Dihedral         = 569.815489
REMARK   1 1,4 Non-bonded   = 499.520121
REMARK   1 1,4 Electrostatic= 4445.065252
REMARK   1
ATOM      1  N   VAL     1       7.357  18.204   5.000   0.058  0.00
ATOM      2  H1  VAL     1       7.744  18.600   5.855   0.227  0.00
ATOM      3  H2  VAL     1       6.358  18.336   4.957   0.227  0.00
ATOM      4  H3  VAL     1       7.576  17.220   4.974   0.227  0.00
ATOM      5  CA  VAL     1       7.948  18.857   3.812  -0.005  0.00
ATOM      6  HA  VAL     1       7.513  18.373   2.927   0.109  0.00
ATOM      7  CB  VAL     1       7.562  20.374   3.761   0.320  0.00
ATOM      8  HB  VAL     1       8.205  20.922   4.460  -0.022  0.00
ATOM      9  CG1 VAL     1       7.734  20.963   2.351  -0.313  0.00
ATOM     10  HG1 VAL     1       7.200  20.370   1.614   0.073  0.00
ATOM     11  HG1 VAL     1       7.348  21.971   2.334   0.073  0.00
ATOM     12  HG1 VAL     1       8.777  21.031   2.074   0.073  0.00
ATOM     13  CG2 VAL     1       6.091  20.612   4.182  -0.313  0.00
ATOM     14  HG2 VAL     1       5.914  20.395   5.230   0.073  0.00
ATOM     15  HG2 VAL     1       5.837  21.655   4.045   0.073  0.00
ATOM     16  HG2 VAL     1       5.401  20.033   3.576   0.073  0.00
ATOM     17  C    VAL     1       9.470  18.591   3.816   0.616  0.00
ATOM     18  O    VAL     1       9.994  18.012   4.791  -0.572  0.00
ATOM     19  N    LEU     2      10.152  18.988   2.739  -0.416  0.00
ATOM     20  H    LEU     2       9.702  19.420   1.936   0.272  0.00
ATOM     21  CA   LEU     2      11.603  19.008   2.683  -0.052  0.00
ATOM     22  HA   LEU     2      11.983  18.097   3.120   0.092  0.00
ATOM     23  CB   LEU     2      12.095  19.097   1.232  -0.110  0.00
ATOM     24  HB2  LEU     2      11.708  20.020   0.810   0.046  0.00
…..
…..
ATOM   2114  CD2  TYR   140      -4.256   9.053 -10.416  -0.191  0.00
ATOM   2115  HD2  TYR   140      -5.071   8.446 -10.050   0.170  0.00
ATOM   2116  C    TYR   140      -7.480  12.287 -10.110   0.597  0.00
ATOM   2117  O    TYR   140      -8.121  11.618 -10.920  -0.568  0.00
ATOM   2118  N    ARG   141      -8.048  12.955  -9.114  -0.348  0.00
ATOM   2119  H    ARG   141      -7.526  13.520  -8.446   0.276  0.00
ATOM   2120  CA   ARG   141      -9.462  13.123  -8.845  -0.307  0.00
ATOM   2121  HA   ARG   141      -9.978  13.465  -9.741   0.145  0.00
ATOM   2122  CB   ARG   141     -10.109  11.835  -8.298  -0.037  0.00
ATOM   2123  HB2  ARG   141     -11.111  12.088  -7.947   0.037  0.00
ATOM   2124  HB3  ARG   141     -10.206  11.103  -9.099   0.037  0.00
ATOM   2125  CG   ARG   141      -9.316  11.209  -7.137   0.074  0.00
ATOM   2126  HG2  ARG   141      -8.389  10.775  -7.516   0.018  0.00
ATOM   2127  HG3  ARG   141      -9.057  11.977  -6.410   0.018  0.00
ATOM   2128  CD   ARG   141     -10.113  10.122  -6.411   0.111  0.00
ATOM   2129  HD2  ARG   141     -11.122  10.491  -6.222   0.047  0.00
ATOM   2130  HD3  ARG   141     -10.167   9.231  -7.040   0.047  0.00
ATOM   2131  NE   ARG   141      -9.476   9.806  -5.122  -0.556  0.00
ATOM   2132  HE   ARG   141      -8.628  10.338  -4.986   0.348  0.00
ATOM   2133  CZ   ARG   141      -9.989   9.061  -4.137   0.837  0.00
ATOM   2134  NH1  ARG   141     -11.125   8.390  -4.322  -0.874  0.00
ATOM   2135  HH1  ARG   141     -11.567   7.834  -3.606   0.449  0.00
ATOM   2136  HH1  ARG   141     -11.600   8.467  -5.211   0.449  0.00
ATOM   2137  NH2  ARG   141      -9.357   8.998  -2.966  -0.874  0.00
ATOM   2138  HH2  ARG   141      -9.719   8.469  -2.187   0.449  0.00
ATOM   2139  HH2  ARG   141      -8.518   9.540  -2.806   0.449  0.00
ATOM   2140  C    ARG   141      -9.530  14.235  -7.814   0.856  0.00
ATOM   2141  O    ARG   141      -8.516  14.373  -7.084  -0.826  0.00
ATOM   2142  OXT  ARG   141     -10.586  14.879  -7.753  -0.826  0.00
```

Parameters:

	Input
PDB structure	Input filename of protein structure (file in PDB format) (http://www.umass.edu/microbio/rasmol/pdb.htm).
Protein chain ID	Protein chain ID.
	Output
Result	Name of the output file.

Net-SSPredict

Program for secondary structure prediction.

Neural nets based on profile of psiBLAST comparison of the query sequence with NR database.

!Attention! This program uses SoftBerry web service and requires the computer should be connected to the internet.

Example:

```
>T0388
Length=174

 PredSS                   bbbbb       aa     bbbbbbbb      aaa
 AA seq     ENLYFQSMINSFYAFEVKDAKGRTVSLEKYKGKVSLVVNVASDCQLTDRN
 ProbA      00242002220000000000000000552110000000000110000766
 ProbB      00002200000334888851103452000100499999985010000000

 PredSS     aaaaaaaaaaaa   bbbbbbb           aaaaaaaaaa    bbb
 AA seq     YLGLKELHKEFGPSHFSVLAFPCNQFGESEPRPSKEVESFARKNYGVTFP
 ProbA      77999999998520000000000121301000089899999971100000
 ProbB      00000000000000389999873100000000000000000000104879

 PredSS     bb           aaaaaaaa          bbbbb    bbbbbb
 AA seq     IFHKIKILGSEGEPAFRFLVDSSKKEPRWNFWKYLVNPEGQVVKFWRPEE
 ProbA      00100000010115888787643000000000000000000000000000
 ProbB      86453442200000000000000000133438988920008999983000

 PredSS      aaaaaaaaaaaaaaaaaa
 AA seq     PIEVIRPDIAALVRQVIIKKKEDL
 ProbA      055688999999999997743000
 ProbB      000000000000000000000000

>T0388
Length=174
   1 E C 0 0
   2 N C 0 0
   3 L C 2 0
   4 Y C 4 0
   5 F C 2 2
   6 Q C 0 2
   7 S C 0 0
   8 M C 2 0
   9 I C 2 0
  10 N C 2 0
  11 S C 0 0
  12 F C 0 3
  13 Y C 0 3
  14 A C 0 4
  15 F B 0 8
  16 E B 0 8
  17 V B 0 8
  18 K B 0 8
  19 D B 0 5
  20 A C 0 1
  21 K C 0 1
  22 G C 0 0
  23 R C 0 3
  24 T C 0 4
  25 V C 0 5
  26 S C 0 2
  27 L A 5 0
  28 E A 5 0
  29 K C 2 0
  30 Y C 1 1
  31 K C 1 0
  32 G C 0 0
  33 K C 0 4
  34 V B 0 9
  35 S B 0 9
  36 L B 0 9
  37 V B 0 9
  38 V B 0 9
  39 N B 0 9
  40 V B 0 8
  41 A B 0 5
  42 S C 1 0
  43 D C 1 1
  44 C C 0 0
  45 Q C 0 0
  46 L C 0 0
  47 T C 0 0
  48 D A 7 0
  49 R A 6 0
  50 N A 6 0
  51 Y A 7 0
  52 L A 7 0
  53 G A 9 0
  54 L A 9 0
  55 K A 9 0
  56 E A 9 0
  57 L A 9 0
  58 H A 9 0
  59 K A 9 0
  60 E A 9 0
  61 F A 8 0
  62 G A 5 0
  63 P C 2 0
  64 S C 0 0
  65 H C 0 3
  66 F B 0 8
  67 S B 0 9
  68 V B 0 9
  69 L B 0 9
  70 A B 0 9
  71 F B 0 8
  72 P B 0 7
  73 C C 0 3
  74 N C 1 1
  75 Q C 2 0
  76 F C 1 0
  77 G C 3 0
  78 E C 0 0
  79 S C 1 0
  80 E C 0 0
  81 P C 0 0
  82 R C 0 0
  83 P C 0 0
  84 S A 8 0
  85 K A 9 0
  86 E A 8 0
  87 V A 9 0
  88 E A 9 0
  89 S A 9 0
  90 F A 9 0
  91 A A 9 0
  92 R A 9 0
  93 K A 7 0
  94 N C 1 0
  95 Y C 1 1
  96 G C 0 0
  97 V C 0 4
  98 T B 0 8
  99 F B 0 7
 100 P B 0 9
 101 I B 0 8
 102 F B 0 6
```

```
103 H C 1 4
104 K C 0 5
105 I C 0 3
106 K C 0 4
107 I C 0 4
108 L C 0 2
109 G C 0 2
110 S C 1 0
111 E C 0 0
112 G C 1 0
113 E C 1 0
114 P A 5 0
115 A A 8 0
116 F A 8 0
117 R A 8 0
118 F A 7 0
119 L A 8 0
120 V A 7 0
121 D A 6 0
122 S C 4 0
123 S C 3 0
124 K C 0 0
125 K C 0 0
126 E C 0 0
127 P C 0 1
128 R C 0 3
129 W C 0 3
130 N C 0 4
131 F C 0 3
132 W B 0 8
133 K B 0 9
134 Y B 0 8
135 L B 0 8
136 V B 0 9
137 N C 0 2
138 P C 0 0
139 E C 0 0
140 G C 0 0
141 Q B 0 8
142 V B 0 9
143 V B 0 9
144 K B 0 9
145 F B 0 9
146 W B 0 8
147 R C 0 3
148 P C 0 0
149 E C 0 0
150 E C 0 0
151 P C 0 0
152 I A 5 0
153 E A 5 0
154 V A 6 0
155 I A 8 0
156 R A 8 0
157 P A 9 0
158 D A 9 0
159 I A 9 0
160 A A 9 0
161 A A 9 0
162 L A 9 0
163 V A 9 0
164 R A 9 0
165 Q A 9 0
166 V A 9 0
```

```
167 I A 9 0
168 I A 7 0
169 K A 7 0
170 K C 4 0
171 K C 3 0
172 E C 0 0
173 D C 0 0
174 L C 0 0
```

Input	
Sequence	Name of input file with protein sequence in FASTA-format.
Output	
Vertical Prediction	Name of the output file with Vertical Prediction.
Horisontal Prediction	Name of the output file with Horisontal Prediction.

NNSSP

Prediction of protein secondary sturcture by combining nearest-neighbor algorithms and multiply sequence alignments

Method description:

Yi and Lander (*) developed a neural-network and nearest-neighbor method with a scoring system that combined a sequence similarity matrix with the local structural environment scoring scheme of Bowie et al.(**) for predicting protein secondary structure. We have improved their scoring system by taking into consideration N- and C-terminal positions of a-helices and b-strands and also b-turns as distinctive types of secondary structure. Another improvement, which also significantly decrease the time of computation, is performed by restricting a data base with a smaller subset of proteins which are similar with a query sequence. Using multiple sequence alignments rather than single sequences and a simple jury decision method we achieved an over all three-state accuracy of 72.2%, which is better than that observed for the most accurate multilayered neural network approach, tested on the same data set of 126 non-homologous protein chains.

Input sequence for this program should be in fasta format with 80 or less sequence letters per line.

(*) Yi T-M., Lander E.S. (1993) Protein secondary structure prediction using nearest-neighbor methods. J.Mol.Biol.,232:1117-1129.

(**) Bowie J.U., Luthy R., Eisenberg D. (1991) A method to identify protein sequences that fold into a known three-dimensional structure. Science, 253, 164-170.)

Accuracy:

Overall 3-states (a, b, c) prediction gives ~67.6% correctly predic- ted residues on 126 non-homologous proteins using the jack-knife test procedure. Using multiple sequence alignments instead of single sequences increases prediction accuracy up to 72.2%.

SEE ALSO **"SSP"** program.

Example of NNSSP output: This output contains probabilities (Pa and Pb) of a and b structures in 0-9 scale. Probability of c is approximately 10 - Pa - Pb.

```
ADENYLATE KINASE ISOENZYME-3, /GTP:AMP$
 L=  214 SS content: a-  0.43 b=  0.05 c=  0.52
                    10        20        30        40        50
 PredSS     aaaaaaa           aaaaaa         aaaaaaaa       aa
 AA seq     RLLRAIMGAPGSGKGTVSSRITKHFELKHLSSGDLLRDNMLRGTEIGVLA
 Prob a     99888651000001112244545422211111346775554221332335
 Prob b     00001221000001134422321222233221001110010101134443
                    60        70        80        90       100
 PredSS     aaaa        aaaaaaaaaaaaaaaa             aaaaaaaaa
 AA seq     KTFIDQGKLIPDDVMTRLVLHELKNLTQYNWLLDGFPRTLPQAEALDRAY
 Prob a     54543201110346789888877545553334210001113588888875
 Prob b     22221001210001111000000000111233410101110000000011
```

```
                  110       120       130       140       150
PredSS         bb     aaaaaaaa   bb      bbbb
AA seq     QIDTVINLNVPFEVIKQRLTARWIHPGSGRVYNIEFNPPKTMGIDDLTGE
Prob a     32111111111466766643321110001100000000000111111111
Prob b     12135643321222110122245531001478764210013333211101
                  160       170       180       190       200
PredSS               aaaaaaaaaaaaaaaaaaaaaaa   bbb          a
AA seq     PLVQREDDRPETVVKRLKAYEAQTEPVLEYYRKKGVLETFSGTETNKIWP
Prob a     23433211146788999997765577888886621121111111123335
Prob b     12321000001110000000000000000000101365542111111221
                  210
PredSS     aaaaaaa
AA seq     HVYAFLQTKLPQRS
Prob a     46687764210111
Prob b     22211110110001
```

Reference:
Salamov A.A., Solovyev V.V.
Prediction of protein secondary sturcture by combining nearest-neighbor algorithms and multiply sequence alignments. J.Mol.Biol.,1995, 247, 11-15.

Parameters:

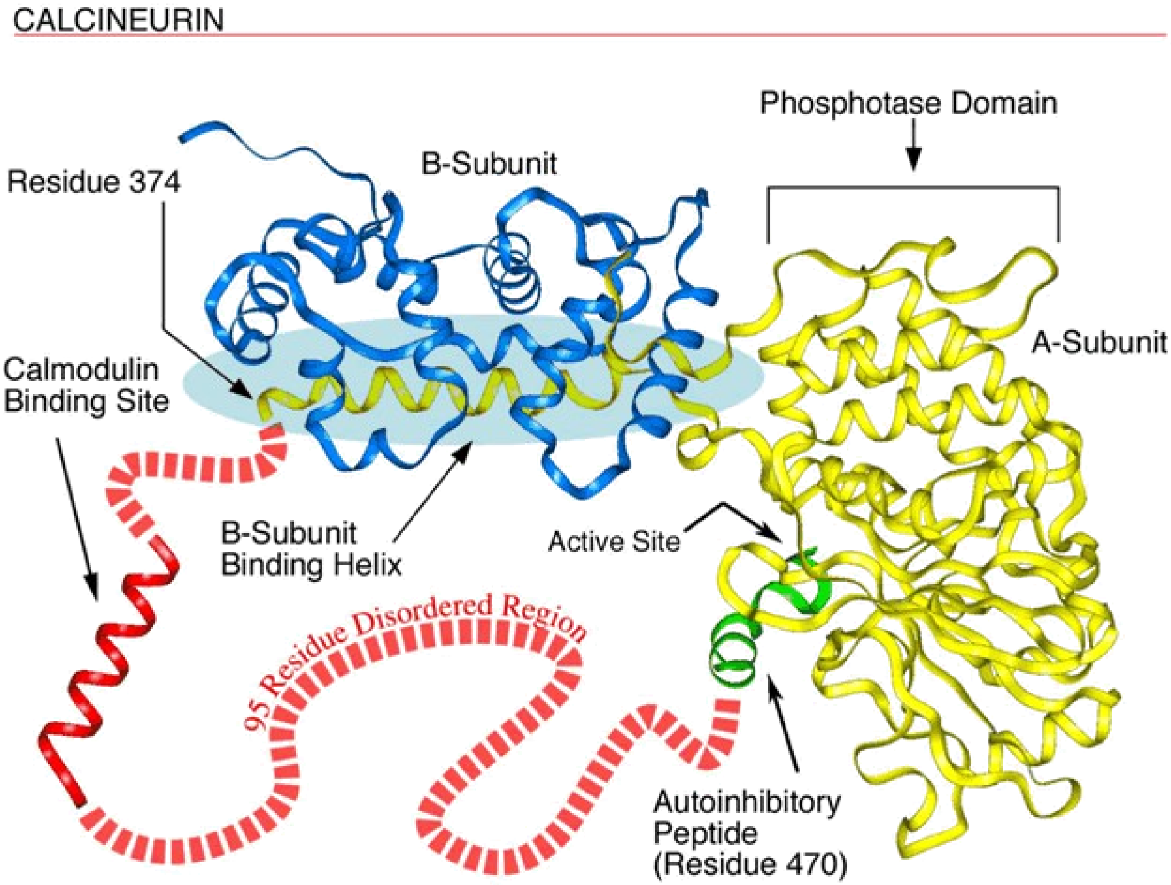
	Input
Sequence	Input file with a sequence. Input sequence for this program should be in fasta format with 80 or less sequence letters per line.
	Output
Result	Name of the output file.

PDisorder

PDisorder is the program for predicting ordered and disordered regions in protein sequences. Minimum required sequence length is 40.

It is increasingly evident that intrinsically unstructured protein regions play key roles in cell-signaling, regulation and cancer (Iakoucheva *et al., J. Mol. Biol.* (2002) 323, 573–584), which makes them extremely useful for discovery of anticancer drugs. Requirement of intrinsic structural disorder is shown for many protein functions - see, for instance, Dunker *et al.*, *Biochemistry* (2002) 41 (21), 6573 -6582.

The figure below shows disorderly region in Calcineurin (reproduced from ORNL Human Genome News (http://www.ornl.gov/TechResources/Human_Genome/publicat/hgn/v12n1/13trinity.html)), see output example below for prediction of its disorder region.



Combination of Neural Network, Linear Discriminant Function and acute Smoothing Procedure is used for recognition of disordered and ordered regions in proteins.

Two sets of significant attributes: one for **Neural Network**, and another one for **Linear Discriminant Function** are selected using automatic LDA procedure, as well as approach based on calculations of **chances to be in disordered or ordered regions**.

Three windowing procedures are used, called **left, right and intermediate**. For all windows, attributes are calculated over **31** residues.

Example of PDisorder output:

```
Prediction of disordered regions in proteins. Softberry Inc.
>gi|1352677|sp|P48457|P2B_EMENI  Ser/thr  protein  phosphatase  2B  catalytic
subunit
Calmodulin-dependent calcineurin A subunit)
                    10        20        30        40
 Pred_od    ooooooooo    ddd ooooooooooooooooooooooooooooooooo
 AA seq     MEDGTQVSTLERVVKEVQAPALNKPSDDQFWDPEEPTKPNLQFLKQHFYR
 Prob_o     66666665655663335777766565589767999999999999997999
                    60        70        80        90
 Pred_od    oooooooooooooooooooooooooooooooooooooooooooooooooo
 AA seq     EGRLTEDQALWIIQAGTQILKSEPNLLEMDAPITVCGDVHGQYYDLMKLF
 Prob_o     99999999999999999999999999999999999999999999999999
                   110       120       130       140
 Pred_od    oooooooooooooooooooooooooooooooooooooooooooooooooo
 AA seq     EVGGDPAETRYLFLGDYVDRGYFSIECVLYLWALKIWYPNTLWLLRGNHE
 Prob_o     99999999999999999999999999999999999999999999999999
                   160       170       180       190
 Pred_od    oooooooooooooooooooooooooooooooooooooooooooooooooo
 AA seq     CRHLTDYFTFKLECKHKYSERIYEACIESFCALPLAAVMNKQFLCIHGGL
 Prob_o     99999999999999999999999999999999999997555556887888
                   210       220       230       240
 Pred_od    oooooooooooooooooooooooooooooooooooooooooooooooooo
 AA seq     SPELHTLEDIKSIDRFREPPTHGLMCDILWADPLEDFGQEKTGDYFIHNS
 Prob_o     78775555553563478776666666678689999999999999999999
                   260       270       280       290
 Pred_od    oooooooooooooooooooooooooooooooooooooooooooooooooo
 AA seq     VRGCSYFFSYPAACAFLEKNNLLSVIRAHEAQDAGYRMYRKTRTTGFPSV
 Prob_o     99999999999999999999999999999999999999999999999999
                   310       320       330       340
 Pred_od    oooooooooooooooooooooooooooooooooooooooooooooooooo
 AA seq     MTIFSAPNYLDVYNNKAAVLKYENNVMNIRQFNCTPHPYWLPNFMDVFTW
 Prob_o     99999999999999999999999999999999999999999999999999
                   360       370       380       390
 Pred_od    ooooooooooo           dddddddddddddddddddddddddddd
 AA seq     SLPFVGEKITDIVIAILNTCSKEELEDETPSTISPAEPSPPMPMDTVDTE
 Prob_o     99999976656555554444441100000000000000000000000000
                   410       420       430       440
 Pred_od    dddddddddddddddddddddddddddddddddddddddddddddddddd
 AA seq     STEFKRRAIKNKILAIGRLSRVFQVLREESERVTELKTAAGGRLPAGTLM
 Prob_o     00000000000100000000001223333444444333422232555555
                   460       470       480       490
 Pred_od    dddddddddddddddddddddddddddddddddddddddddddddddddd
 AA seq     LGAEGIKQAITNFEDARKVDLQNERLPPSHDEVVRRSEEERRIALDRAQH
 Prob_o     55555433255544555565443400000231112100000000000001
                   510       520
 Pred_od    dddddddddddddddddddddddddddddd
 AA seq     EADNDTGLATVARRISMVRRIRKIPSTTRR
 Prob_o     020000022332232444444444443343

sequences=1 disordered=161 ordered=353 unknown=16
```

Here line **Pred_od** shows ordered (o) and disordered (d) regions. Blanks denote undefined-state stretches, usually at boundaries of disordered regions.

Line **Prob_o** shows raw probability on a scale of 0 to 9 for each amino acid residue to be in ordered region.

The line at the end of the output shows total number of sequence residues in each state: disordered, ordered and unknown.

Accuracy estimations:

One of accuracy tests was made on PONDR data and in comparison with PONDR.

Black and **blue** - PONDR's data, **green** - our descriptions, **red** - PDisorder results.

PONDR and PDisorder accuracies

Predictor	False Negative (dis_ALL) - 124 sequences >31 in lengths, 17181 positions (false, true)		False Positive (O_PDB_S25) - 1081 sequences >31 in lengths, 220743 positions (false, true)		5-cross Validation	Unknown (for both sets)
VL-XT	40%	-	22%	-	75 - 83%	-
XL1	62%	-	19%	-	73 ± 4%	-
CaN	39%	-	34%	-	83 ± 5%	-
PDisorder	20.3%	78.3%	4.7%	94.4%	-	0.7%

Parameters:

Input	
Sequences set	File with sequences in FASTA format.
Output	
Result	Name of the output file.

PSSFinder

PSSFinder predicts the secondary structure of queried protein using the information on homology from the database.

Parameters:

	Input
Sequences set	Name of the input FASTA protein file (single or set).
	Output
Result	Name of the output file.
CHE-style	nly secondary structure in C(coil) H (Helix) E(b-strand) alphabet.
String length	Count of symbols by line.
	Options
Fine mode (very slow)	Fine mode - near the 1000 times slowly.

SSEnvID

Protein secondary structure and environment assignment from atomic coordinates

SSEnvID is a program to recognize secondary structural elements in proteins from their atomic coordinates. It performs the same task as DSSP by Kabsch and Sander (1983) or STRIDE by Frishman & Argos (1995) with analyzing both hydrogen bond and mainchain dihedral angles, as well some probabilistic measures. SSEnvID also computes accessible surface area, polarity and environment classes as defined by Bowie, Luthy, Eisenberg (1991). SSEnvID's new feature is the probability (quality) of secondary structure assignment for each amino acids.

SSEnvID computes 3D protein characteristics which are used in structure prediction by measuring the compatibility between protein sequences and known protein structures.

SSEnvID output:

```
  SSEnvID - Protein secondary structure and environment assignment
             from atomic coordinates (Softberry Inc., 2001)

  Ch   - Chain
  ResN - PDB resnumber
  Nam  - Amino acid sequence in three letter code
  Ab   - Area Buried
  Fp   - Fraction Polar
  SS   - Secondary structure assignment (E-beta sheet, H,G,I-helices, T-turn)
  PDBSS- Original PDB secondary structure assignment (if provided)
  Env  - Side-Chain Environment Class
  PrHel- Probability of helix
  PrBet- Probability of beta bridge

Ch     ResN  Nam    Ab      Fp   SS  PDBSS   Env    PrHel   PrBet

  A      1  VAL   79.1   0.35  C     C    P1       0.00    0.00
  A      2  ALA   26.2   0.60  C     C    E        0.00    0.09
  A      3  ILE  157.0   0.23  E     C    B1       0.13    0.88
  A      4  LYS  105.5   0.72  E     C    P2       0.13    0.88
  A      5  MET  172.0   0.30  E     C    B1       0.13    0.88
  A      6  GLY   40.0   0.37  C     C    E        0.13    0.16
  A      7  ALA   64.5   0.47  C     C    P1       0.13    0.00
  A      8  ASP   54.5   0.77  T     C    P2       0.08    0.00
  A      9  ASN   36.7   0.57  T     C    E        0.08    0.00
  A     10  GLY   14.0   0.53  C     C    E        0.07    0.00
  A     11  MET   33.1   0.80  C     C    E        0.13    0.00
```

```
A    12  LEU   97.5   0.49  C    C   P1      0.13    0.01
A    13  ALA   53.7   0.47  C    C   P1      0.13    0.07
A    14  PHE  188.1   0.34  C    C   B2      0.13    0.88
A    15  GLU   96.0   0.54  C    C   P1      0.13    0.88
A    16  PRO   66.5   0.56  C    C   P1      0.13    0.00
A    17  SER   34.9   0.81  C    C   E       0.13    0.00
A    18  THR   57.7   0.63  E    E   P2      0.13    0.86
A    19  ILE  139.9   0.29  E    E   B1      0.13    0.86
A    20  GLU   87.9   0.51  E    E   P1      0.13    0.88
A    21  ILE  157.0   0.35  E    E   B2      0.13    0.88
A    22  GLN   45.2   0.80  C    E   P2      0.16    0.00
A    23  ALA   47.2   0.56  T    C   P1      0.16    0.16
A    24  GLY   21.5   0.61  T    C   E       0.16    0.00
A    25  ASP   70.7   0.46  C    C   P1      0.16    0.30
A    26  THR   63.0   0.71  E    E   P2      0.13    0.88
A    27  VAL  129.9   0.24  E    E   B1      0.13    0.88
A    28  GLN   95.7   0.50  E    E   P1      0.13    0.88
A    29  TRP  234.0   0.16  E    E   B1      0.13    0.90
A    30  VAL  112.0   0.42  E    E   P1      0.13    0.90
A    31  ASN  122.7   0.41  E    E   B2      0.26    0.88
A    32  ASN   90.0   0.54  C    C   P1      0.26    0.00
A    33  LYS   91.2   0.71  C    C   P2      0.26    0.01
A    34  LEU   38.7   0.66  C    C   E       0.13    0.00
A    35  ALA   56.4   0.64  C    C   P2      0.13    0.01
A    36  PRO   70.4   0.47  C    C   P1      0.13    0.00
A    37  HIS  175.0   0.30  E    C   B1      0.13    0.90
A    38  ASN  117.8   0.37  E    C   B2      0.13    0.17
A    39  VAL  130.0   0.18  E    C   B1      0.13    0.88
A    40  VAL  111.6   0.48  E    C   P1      0.13    0.87
A    41  VAL  129.2   0.24  E    C   B1      0.13    0.87
A    42  GLU   51.1   0.68  T    C   P2      0.08    0.17
A    49  GLY    0.0   0.77  T    C   E       0.08    0.09
A    52  GLN  104.9   0.50  C    C   P1      0.22    0.30
A    53  PRO    0.0   0.86  G    H   E       0.96    0.00
A    54  GLU   50.1   0.69  G    H   P2      0.96    0.00
A    55  LEU  144.4   0.34  G    H   B2      0.96    0.00
A    56  SER   81.2   0.40  C    C   P1      0.07    0.00
A    57  HIS  111.3   0.53  E    C   P1      0.13    0.88
A    58  LYS   10.1   0.81  E    C   E       0.13    0.00
A    59  ASP    0.0   0.82  E    C   E       0.13    0.00
A    62  LEU   83.4   0.49  E    C   P1      0.13    0.17
A    63  ALA   70.5   0.46  E    C   P1      0.26    0.90
A    64  PHE   20.5   0.67  C    C   E       0.26    0.01
A    65  SER   22.2   0.74  C    C   E       0.26    0.00
A    66  PRO   10.6   0.83  T    C   E       0.34    0.17
A    67  GLY   21.1   0.56  T    C   E       0.34    0.00
A    68  GLU  102.2   0.56  C    C   P1      0.34    0.09
A    69  THR   73.7   0.54  E    E   P1      0.13    0.90
A    70  PHE  165.9   0.41  E    E   B2      0.13    0.90
A    71  GLU   83.4   0.56  E    E   P1      0.13    0.88
A    72  ALA   58.9   0.46  E    E   P1      0.13    0.88
A    73  THR   57.1   0.67  E    E   P2      0.13    0.88
A    74  PHE  188.9   0.22  C    C   B1      0.13    0.30
A    75  SER   27.9   0.59  C    C   E       0.13    0.00
A    76  GLU    0.0   0.86  C    C   E       0.13    0.00
.......
```

Parameters:

Input	
PDB structure	Input filename of protein structure (file in PDB format) (http://www.umass.edu/microbio/rasmol/pdb.htm).
Chain	Protein chain ID.
Output	

Result	Name of the output file.

SSP

Prediction of a-helix and b-strand segments of globular proteins

Method description:

Our segment-oriented method is designed to locate secondary structure elements and uses linear discriminant analysis to assign segments of a given amino acid sequence to a particular type of secondary structure, by taking into account the amino acid composition of internal parts of segments as well as their terminal and adjacent regions. Four linear discriminant functions were constructed for recognition of short and long a-helix and b-strand segments, respectively. These functions combine 3 characteristics: hydrophobic moment, segment singlet and pair preferences to an a-helix or b-strand. To improve the prediction accuracy of the method, a simple version which treats multiple sequence alignments that are used as input in place of single sequences has been developed.

Accuracy:

Overall 3-states (a, b, c) prediction gives ~65.1% correctly predicted residues on 126 non-homologous proteins using the jack-knife test procedure (The accuracy is good if you have no homologous sequences to apply Sander et al. method (Rost,Sander, Mol.Biol,1993,232,584-599) that has about 71% accuracy with using these sequences and about 61% without them). Analysis of the prediction results shows high prediction accuracy of long secondary structure segments (~89% of a- helices of lengths greater than 8 and ~71% of b-strands of lengths greater than 6 are located with probability of correct prediction 0.82 and 0.78 respectively). Using mean values of discriminant functions over the aligned sequences of homologous proteins, we achieved a prediction accuracy of 68.2%. Our variant of nearest-neighbor algorithm with using multiply sequence alignments of homologous proteins has 72% accuracy and 67.6% accuracy without homologous proteins.

SEE ALSO **NNSSP** program.

Loading File Format:

(a) For single sequence you must load file in the following format:

First Line - Sequence name,

Second line - number 1 in format I5,

Third and subsequent lines - amino acid sequence.

Sequence length must be less than 2000 amino acids! Restrict the line length to 75 characters. You can use small letters for Cys bridges, if you want.

Example:

```
ADENYLATE KINASE
    1
RLLRAIMGAPGSGKGTVSSRITKHFELKHLSSGDLLRDNMLRGTEIGVLA
KTFIDQGKLIPDDVMTRLVLHELKNLTQYNWLLDGFPRTLPQAEALDRAY
QIDTVINLNVPFEVIKQRLTARWIHPGSGRVYNIEFNPPKTMGIDDLTGE
PLVQREDDRPETVVK............
```

(b) For multiple aligned sequences:

First Line - Sequence name,

Second line - number of aligned sequences and length of protein,

Third line - empty or numbers of aligned aminoacid sequence,

Subsequent lines - aligned amino acid sequences in format 60a1.

Parts of aligned sequences must be separated by empty line or line with numbers. The number of aligned sequences must be less than 250. Alignment MUST be without gaps in the first (query) sequence!

Example:

```
ACTINOXANTHIN
5 107
```

```
        10        20        30        40        50        60
APAFSVSPASGASDGQSVSVSVAAAGETYYIAQaAPVGGQDAaNPATATSFTTDASGAAS
APAFSVSPASGLSDGQSVSVSGAAAGETYYIAQCAPVGGQDACNPATATSFTTDASGAAS
APTATVTPSSGLSDGTVVKVAGAgaGTAYDVGQCAWVdgVLACNPADFSSVTADANGSAS
APGVTVTPATGLSNGQTVTVSATgpGTVYHVGQCAVvpGVIGCDATTSTDVTADAAGKIT
ATPKSSSGGAGASTGSGTSSAAVTSgaASSAQQSGLQGATGAGGGSSSTPGTQPGSGAGG
        70        80        90       100
FSFTVRKSYAGQTPSGTPVGSVDbATDAbNLGAGNSGLNLGHVALTF
FSFV-RKSYAGZTPSGTPVGSVDCATDACNLGAGNSGLNLGHVALTF

TSLTVRRSFEGFLFDGTRWGTVDCTTAACQVGLSDAAGNGpgVAISF
AQLKVHSSFQAVvaNGTPWGTVNCKVVSCSAGLGSDSGEGAAQAITF
AIAARPVSAMGGtpPHTVPGSTNTTTTAMAGGVGGPgaNPNAAALM-
```

Example of SSP output:

```
   ADENYLATE KINASE
                    10        20        30        40        50
   pred A:    aaaaaaaaa          aaaaaaaaa     aaaaaaaaa     aaa
   AA         N  4.1  C          N  2.2  C     N  4.4  C     N
   pred B:                  bbbb
   BB                       N2 C
   Predic     aaaaaaaaa     bbbb aaaaaaaaa     aaaaaaaaa     aaa
   a/acid     RLLRAIMGAPGSGKGTVSSRITKHFELKHLSSGDLLRDNMLRGTEIGVLA
                    60        70        80        90       100
   pred A:    aaaaaa       aaaaaaaaaaaaaaaaaaaaaaa     aaaaaaaaa
   AA         2.2  C       N    4.2    CN   2.4  C     N  5.4  C
   pred B:                 bbbbbbb
   BB                      N 2.6 C
   Predic     aaaaaa       aaaaaaaaaaaaaaaaaaaaaaa     aaaaaaaaa
   a/acid     KTFIDQGKLIPDDVMTRLVLHELKNLTQYNWLLDGFPRTLPQAEALDRAY
```

The output of the prediction program presents not only final optimal variant of the secondary structure assignment, but also a set of potential a-helix and b-strand segments that were computed without consideration of their competition. Because the protein secondary structure is finally stabilized during the formation of the tertiary structure, the alternative variants of the a-helix and b-strand segments may be important for methods of tertiary structure prediction.

References:

Solovyev V.V.,Salamov A.A. Method of calculation of discrete secondary structures in globular proteins. Molek. Biol. 25:810-824,1991 (in Russ.)
Solovyev V.V.,Salamov A.A. 1994, Secondary structure prediction based on discriminant analysis. In Computer analysis of Genetic macromolecules. (eds. Kolchanov N.A., Lim H.A.), World Scientific, p.352-364.
Solovyev V.V., Salamov A.A. Predicting a-helix and b-strand segments of globular proteins. CABIOS (1994), V.10,6,661-669

Parameters:

	Input
Sequence	Name of input file with protein sequence in FASTA-format. Sequence length must be less than 2000 amino acids! Restrict the line length to 75 characters. You can use small letters for Cys bridges, if you want.
	Output
Result	Name of the output file.

SSPAL

Prediction of protein secondary structure by using local alignments.

Method is based on comparison of charcteristics, calculated for positions of processing sequence, such as aminoacid exposure to water, submergence of aminoacid residue into molecule body etc, with the same characteristics, obtained from analysis of PDB-files in database.

FASTA formatted sequence or specially prepared alignment (see example) can be used as an input. The number of aligned sequences must be less than 250 !!!

Input sequence for this program should be in fasta format with 80 or less sequence letters per line.

Accuracy
Overall 3-state (a, b, c) prediction gives about 75% correctly predicted residues. THIS ACCURACY IS REACHED WITHOUT USING MULTIPLE ALIGNMENT INPUT when it is higher SEE ALSO "SSP" and "NNSSP" programs.
Output results with probability of prediction:

```
Length=136
                    10        20        30        40        50
 PredSS       aaaaaaaaaaaa     aaaaaaaaaaa aaaa          aaaa
 AA seq     LSADQISTVQASFDKVKGDPVGILYAVFKADPSIMAKFTQFAGKDLESIK
 ProbA      11999999999999111119999999999919999111111111199991
 ProbB      11000000000000111110000000000010000111111111100001
                    60        70        80        90       100
 PredSS       aaaaaaaaaaaaaaaaaaa     aaaaaaaaaaa      aaaaaaa
 AA seq     GTAPFETHANRIVGFFSKIIGELPNIEADVNTFVASHKPRGVTHDQLNNF
 ProbA      11999999999999999999911111999999999991111119999999
 ProbB      11000000000000000000011111000000000001111110000000
                   110       120       130
 PredSS     aaaaaaaaaaa      aaaaaaaaaaaaaaaaaa
 AA seq     RAGFVSYMKAHTDFAGAEAAWGATLDTFFGMIFSKM
 ProbA      999999999991111119999999999999999991
 ProbB      000000000001111110000000000000000001
```

- 1 line - sequence name
- 2 line - number of aligned sequences and length of protein
- 3 and subsequent lines - aligned sequences in format 60a1
- (where 3-d line is empty or with numbers as well as other lines
- which separate parts of aligned sequences)

for example:

```
     ACTINOXANTHIN
         5  107
              10         20        30         40        50         60  (numbers
not
     APAFSVSPASGASDGQSVSVSVAAAGETYYIAQaAPVGGQDAaNPATATSFTTDASGAAS  necessary)
     APAFSVSPASGLSDGQSVSVSGAAAGETYYIAQCAPVGGQDACNPATATSFTTDASGAAS
     APTATVTPSSGLSDGTVVKVAGAgaGTAYDVGQCAWVdgVLACNPADFSSVTADANGSAS
     APGVTVTPATGLSNGQTVTVSATgpGTVYHVGQCAVvpGVIGCDATTSTDVTADAAGKIT
     ATPKSSSGGAGASTGSGTSSAAVTSgaASSAQQSGLQGATGAGGGSSSTPGTQPGSGAGG
             70        80        90       100
     FSFTVRKSYAGQTPSGTPVGSVDbATDAbNLGAGNSGLNLGHVALTF
     FSFV-RKSYAGZTPSGTPVGSVDCATDACNLGAGNSGLNLGHVALTF
     TSLTVRRSFEGFLFDGTRWGTVDCTTAACQVGLSDAAGNGpgVAISF
     AQLKVHSSFQAVvaNGTPWGTVNCKVVSCSAGLGSDSGEGAAQAITF
     AIAARPVSAMGGtpPHTVPGSTNTTTTAMAGGVGGPgaNPNAAALM-
```

```
      (you can use small letters for Cys amino acids, if you want)

      Alignment MUST be without  deletions in the 1-st (query) sequence!!!
```

References:

Salamov A.A., Solovyev V.V. Protein secondary sturcture prediction using local alignments. J.Mol.Biol.1977, 268,1, 31-36.

Salamov A.A., Solovyev V.V. Prediction of protein secondary sturcture by combining nearest-neighbor algorithms and multiply sequence alignments. J.Mol.Biol.1995,247,1,11-15.

Parameters:

Input	
Data	Input file with a sequence in FASTA-format or specially prepared alignment (see example in Help). Input sequence for this program should be in fasta format with 80 or less sequence letters per line.
Output	
Result	Name of the output file.

Proteomics

This is a collaborative project for analysis of mass spectra data with Universal Prediction Limited (UK)(http://www.universal-prediction.com/).

The information contained in mass spectra, in combination with the level of tumor marker serum CA125 useful for early detection of ovarian cancer (Gammerman *et al*.,The Computer Journal, (2008))

MS data processing can be used to solve this task.

First step of analysis is data preprocessing that allow to compare MS from different patients and to identify location of peaks.

The Softberry SMS program package allows to perform these procedures are used to be completed in the following order:

Data resampling;

Data smoothing;

Detection of the baseline and its subtraction from intensity;

Normalization;

Peaks identification.

Once the peaks in different spectra are identified, they can be aligned over each other that allows to reveal the presence of common peaks in these spectra.

MSBaseline

Proteomics-MSBaseline- Softberry Mass Spectra (SMS) processing tools. Baseline detection and subtraction.

This step of data processing is applied for elimination of the systematic artifacts that occur due to matrix and chemicals used in the experiments or as a result of detector overload. It results in background noise that may occur to be significant for some*m*values. The initial step in background noise removal is identification of peaks (local signal maxima that are located far enough from each other). The distance between peaks is determined by the '*Baseline parameter*' value (default= 0.005). This parameter defines the minimal*m*distance, over which the two neighboring peaks 1 and 2 are to be located in the way, when:

$|m_1-m_2|/m_1>$ '*Baseline parameter*'.

After peaks identification, algorithm detects the points with signal minima located in intervals between peaks. These are the base points for calculation of background noise line. Over base points the baseline for all spectrum points is built by interpolation. In case when in some spectrum parts the value of base signal exceeds the original one, the new base points selection from neighboring ones occurs.

The values of base signal intensity are subtracted from the original one. At that, if value of original signal has occurred below zero, it is equated to zero. The result of background subtraction is shown in figure 1.

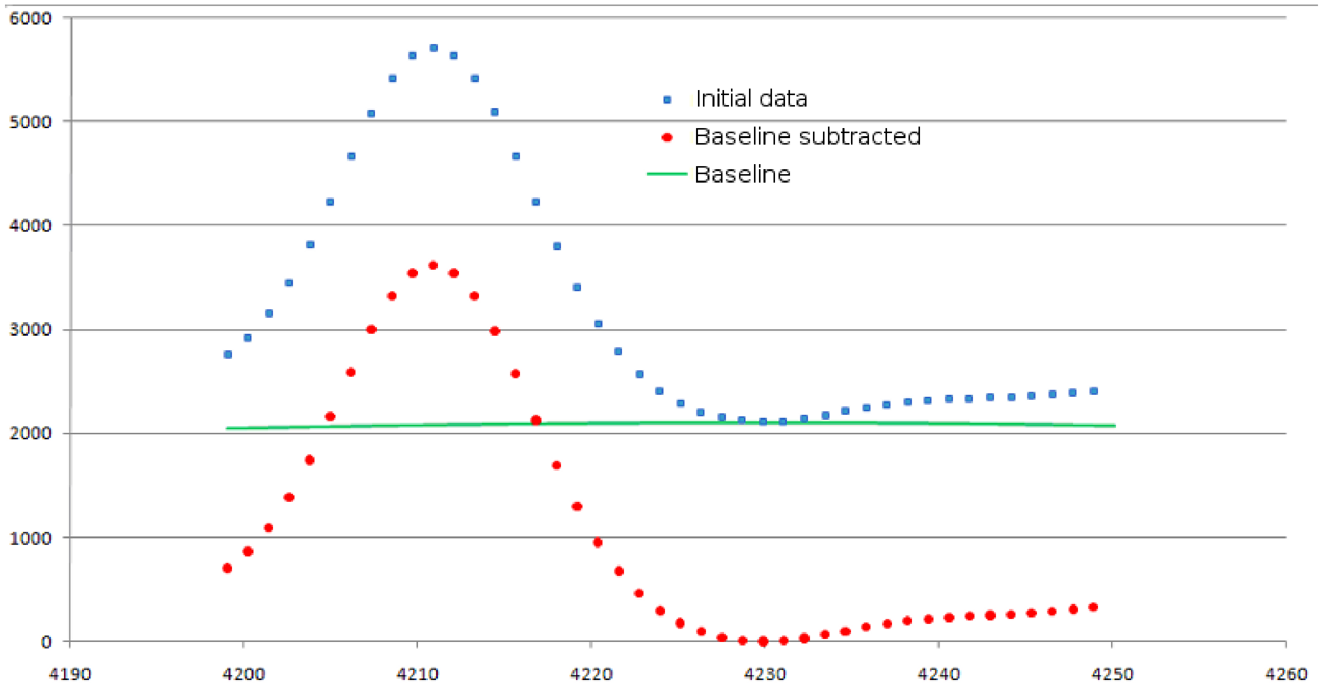


Figure 1. Result of background signal subtraction. Original data are shown as blue squares, modified ones - as red circles. Baseline is shown in green line.

Input: m/z - Intensity data

Output: m/z - Intensity data after baseline subtraction in the same format as input data.

Parameter(s):

Baseline parameter- This parameter specify the minimal mass difference, over which the two neighboring peaks 1 and 2 are to be distinguished for baseline determination. The default value is 0.005.

File format type- This parameter specify file format. SSV-space separated values, CSV - comma separated values, TSV - tab separated values.

Data format.

Mass spectra data represent the sets of following pairs of values: mass to charge relation (m/z, further, for more convenience, it will be referred to as m, mass) and corresponding signal intensity (I). On a spectrum plot, the mass corresponds to X coordinate, and signal intensity- to Y one. A typical spectrum consists of several thousand of such value pairs (points). Data are represented as text files, where for each pair (m_i,I_i) of mass-intensity values the string is assigned, and data in this string are separated by special separator symbol. The SMS package allows several separators types: space (SSV, space separated values, file format), comma (CSV, comma separated values, file format) and tabulation (TSV, tab-separated values, file format). In files with data, the string with comments are allowed; during the file reading these strings are to be skipped. The commentary strings should begin with "#" symbol at the first position. In the figure 2 the example of file with data in CSV format is shown.

```
#M/Z,Intensity
-7.8602611e-005,4.1126194
2.1773576e-007,4.0764203
9.6021472e-005,4.0040221
0.00036601382,4.1186526
0.00081019477,4.0040221
0.0014285643,3.9617898
….
19742.941,4.077895
19745.564,4.0772248
```

```
19748.187,4.0772248
```

Figure 2. Example file with mass spectra data in CSV format.

Parameters:

Input	
Input data file	File with input data.
Input Files Format	This parameter specify the format of input files: Space separated values, Comma separated values, Tab separated values.
Output	
Result	Name of output file
Options	
Baseline parameter	This parameter specify the minimal distance between two neighboring peaks for baseline determination.

MSCalcParamLDA

MSCalcParamLDAsearches for the linear combination of features (the level of additional marker and peak intensities), which allow to distinguish two classes of samples (e.g. cancer and non-cancer samples).

Parameters:

Input	
Peaks Intensity Data File	Text file with results of MSCreateTable program output.
Peaks Group Data File	Text file with results of MSPeakAlign program output for the same mass spectra data set as in table data.
Output	
Result	Name of output file
Options	
Number of best peaks	This parameter specify number of peaks' data (top represented in sample set) to be used for LDA calculation.
Sampling Max Time	This parameter defines maximal time of sampling (in months prior diagnosis) for cancer patients in order to limit samples to be included in the cancer training set. The time of sampling is specified in the Table data in column "time".

MSCalibrate

MSCalibrate - program scaling and shifting operations on the raw mass-spectrum data.
The parameters of transformations estimated using data from calibration spectra. The calibration allows removing systemic noise from hardware.
Parameters:

Input	
Input data file	File with input data.
Calibration Data	Text file with calibration data.**!Note!**Format should be identical to that of Input Data file.
Input Files Format	This parameter specify the format of input files: Space separated values, Comma separated values, Tab separated values.

Output	
Result	Name of output file
Options	
Calibration peaks location	The string should contain the list of comma separated M/Z values for calibration peaks.
Minimal mass separation	This parameter specifies the minimal distance to distinguish neighbouring peaks in the calibration data.
SNR window size	This parameter specifies the window size to determine signal-to-noise ratio.

MSCreateTable

MSCreateTable- program created data table for linear discriminant analysis.

Parameters:

Input	
Input Data Folder	Input Data Folder.
Preprocessed Spectra Data Set = "File"	Text file with the names of preprocessed spectra files.
Input Files Format	This parameter specify the format of input files: Space separated values, Comma separated values, Tab separated values.
Extra Data File	Text file with tab-separated data for each sample from data file list: sample index, time of sampling, patient ID, case (cancer=1, non-cancer=0), log value of CA125 marke
Peak Data File	The name of peak data file (result of MSPeakFind program).
Peaks Group Data File	The name of file with peak group data (output of MSPeakAlign program).
Output	
Result	Name of output file
Options	
Output number of top peaks group	This parameter specify the number of mostly presented peak groups in processed samples for output.

MSNormalization

Proteomics-MSNormalization- Softberry Mass Spectra (SMS) processing tools. Data normalization.

Normalization allows to bring peaks intensity values to a common scale, and thus it becomes possible to compare data from different spectra. The only parameter for current procedure is '*NormalizationConstant*' (default value is 10000).

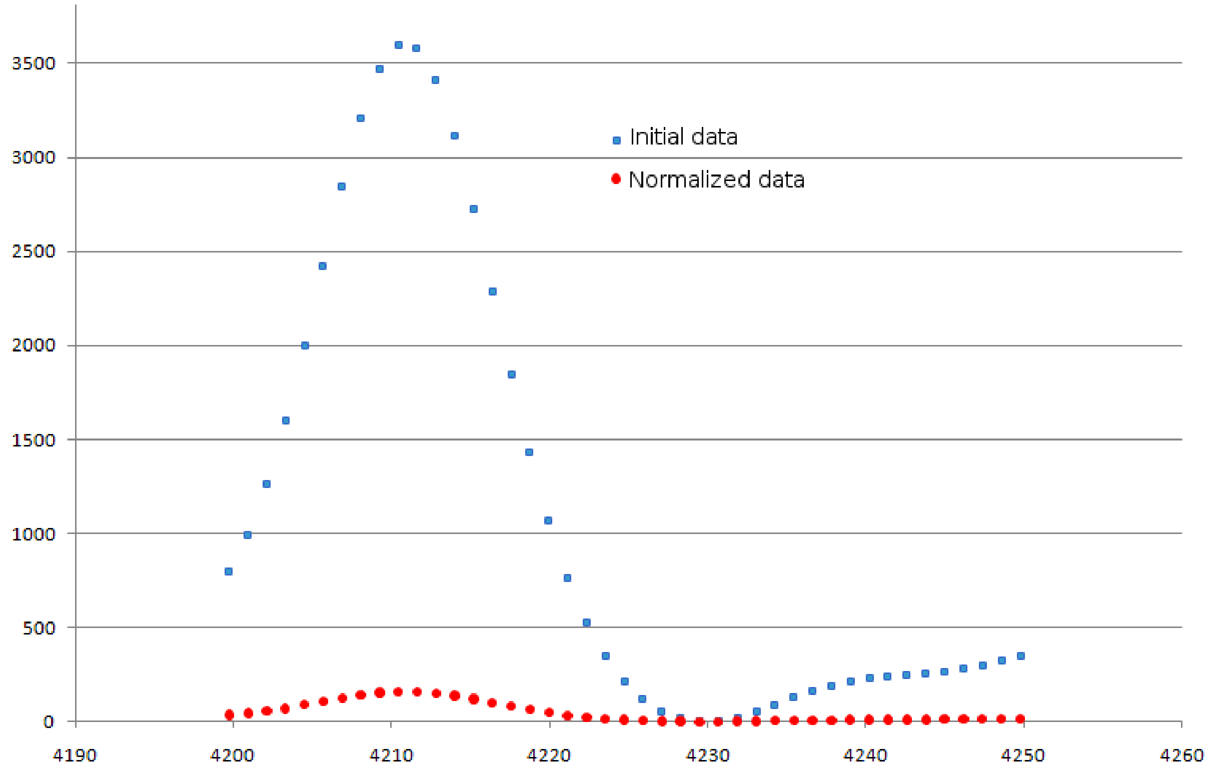


Figure 1. Result of normalization procedure. Original data are shown as blue squares, modified ones - as red circles.

Input:m/z - Intensity data

Output: Normalized m/z - Intensity data in the same format as input data.

Parameter(s):

NormalizationConstant- This parameter specify the normalization constant. The default value is 10000.

File format type- This parameter specify file format. SSV-space separated values, CSV - comma separated values, TSV - tab separated values.

Data format.

Mass spectra data represent the sets of following pairs of values: mass to charge relation (m/z, further, for more convenience, it will be referred to as m, mass) and corresponding signal intensity (I). On a spectrum plot, the mass corresponds to X coordinate, and signal intensity- to Y one. A typical spectrum consists of several thousand of such value pairs (points). Data are represented as text files, where for each pair (m_i,I_i) of mass-intensity values the string is assigned, and data in this string are separated by special separator symbol. The SMS package allows several separators types: space (SSV, space separated values, file format), comma (CSV, comma separated values, file format) and tabulation (TSV, tab-separated values, file format). In files with data, the string with comments are allowed; during the file reading these strings are to be skipped. The commentary strings should begin with "#" symbol at the first position. In the figure 2 the example of file with data in CSV format is shown.

```
#M/Z,Intensity
-7.8602611e-005,4.1126194
2.1773576e-007,4.0764203
9.6021472e-005,4.0040221
0.00036601382,4.1186526
0.00081019477,4.0040221
0.0014285643,3.9617898
….
19742.941,4.077895
```

```
19745.564,4.0772248
19748.187,4.0772248
```

Figure 2. Example file with mass spectra data in CSV format.

Parameters:

Input	
Input data file	File with input data.
Input Files Format	This parameter specify the format of input files: Space separated values, Comma separated values, Tab separated values.
Output	
Result	Name of output file
Options	
Normalization constant	This parameter specifies the normalization constant.

MSPeakAlign

Proteomics-MSPeakAlign- Softberry Mass Spectra (SMS) processing tools. Peaks detection and alignment.

This program finds peaks in several samples and aligns them. For each single spectrum this program performs:

(1) Data resampling;
(2) Data smoothing;
(3) Detection of the baseline and its subtraction from intensity;
(4) Normalization;
(5) Peaks identification.

Once the peaks in different spectra are identified, they can bealignedover each other that allows to reveal the presence of common peaks in these spectra.

Step 1. Data resampling.

The first step in mass spectra processing is data resampling. It allows to discriminate the excessive data and to bring the$m_i$values to common scale. As a result, different spectra will have the same$m$value counts, and, thus, will be comparable. Reduction in number of spectrum points allows to lower the noise and to eliminate excessive data, but, at the same time, to keep the spectrum shape. The common data scale after conversion is located between the minimal and maximal$m$values of spectrum. The number of data that will be resampled from original set is determined by the '*Binning percent*' parameter, that represents the percentage of spectrum points remained after conversion (default value is 25). Example of data resampling is shown in figure 1.

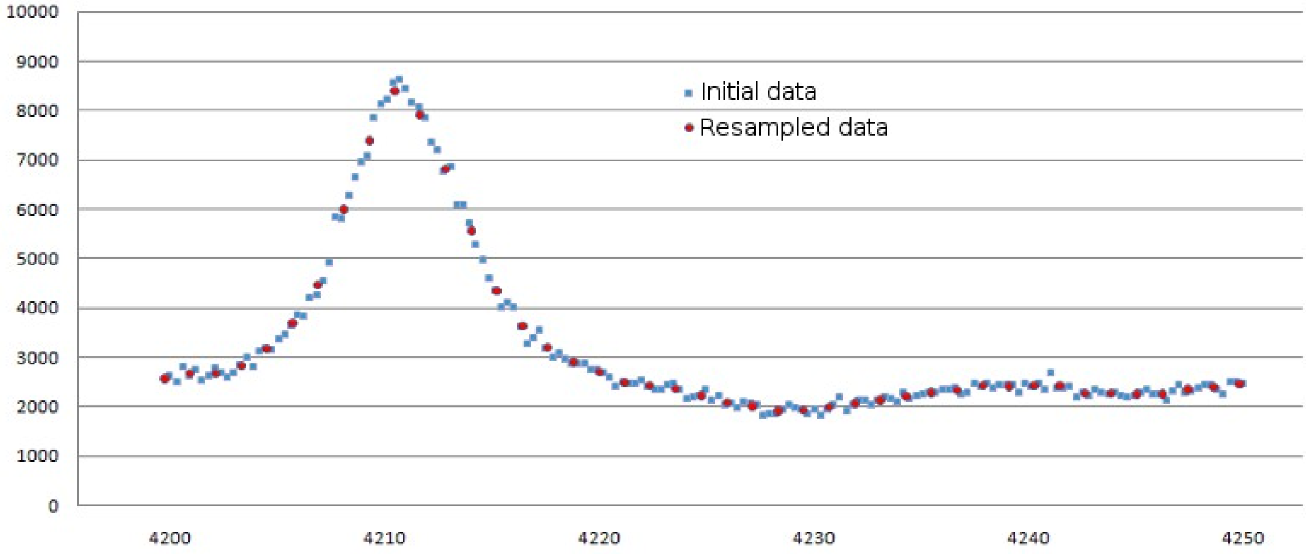


Figure 1. Result of data resampling for small spectrum interval. Original data are shown as blue squares, resampled ones - as red circles. The '*Binning percent*' for this case was set to 25.

Step 2. Smoothing.

Data smoothing procedure is intended for data noise elimination. During the smoothing, the values of intensity for each$m_i$point are being averaged by several neighboring points. The number of such points is determined by the '*SmoothWindowSize*' parameter (default value is 3). The smoothing procedure can be repeated for several times; the number of iterations is determined by the '*SmoothReps*' parameter (default value is 3). Example of data smoothing is shown in the figure 2.

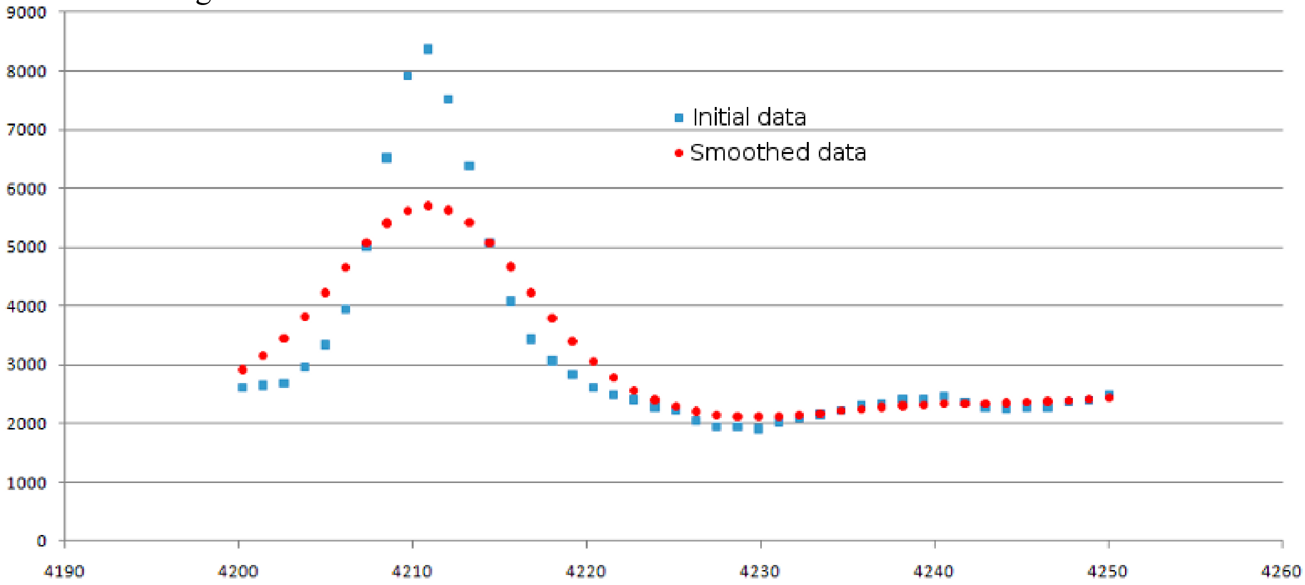


Figure 2. Result of data smoothing. Original data are shown as blue squares, smoothed ones - as red circles. The*SmoothWindowSize*was set to 3 and*SmoothReps*was set to 3.

Step 3. Baseline detection and subtraction.

This step of data processing is applied for elimination of the systematic artifacts that occur due to matrix and chemicals used in the experiments or as a result of detector overload. It results in background noise that may occur to be significant for some*m*values. The initial step in background noise removal is identification of peaks (local signal maxima that are located far enough from each other). The distance between peaks is determined by the '*Baseline*

parameter' value (default= 0.005). This parameter defines the minimal*m*distance, over which the two neighboring peaks 1 and 2 are to be located in the way, when:
$|m_1\text{-}m_2|/m_1>$ '*Baseline parameter*'.

After peaks identification, algorithm detects the points with signal minima located in intervals between peaks. These are the base points for calculation of background noise line. Over base points the baseline for all spectrum points is built by interpolation. In case when in some spectrum parts the value of base signal exceeds the original one, the new base points selection from neighboring ones occurs.

The values of base signal intensity are subtracted from the original one. At that, if value of original signal has occurred below zero, it is equated to zero. The result of background subtraction is shown in figure 3.

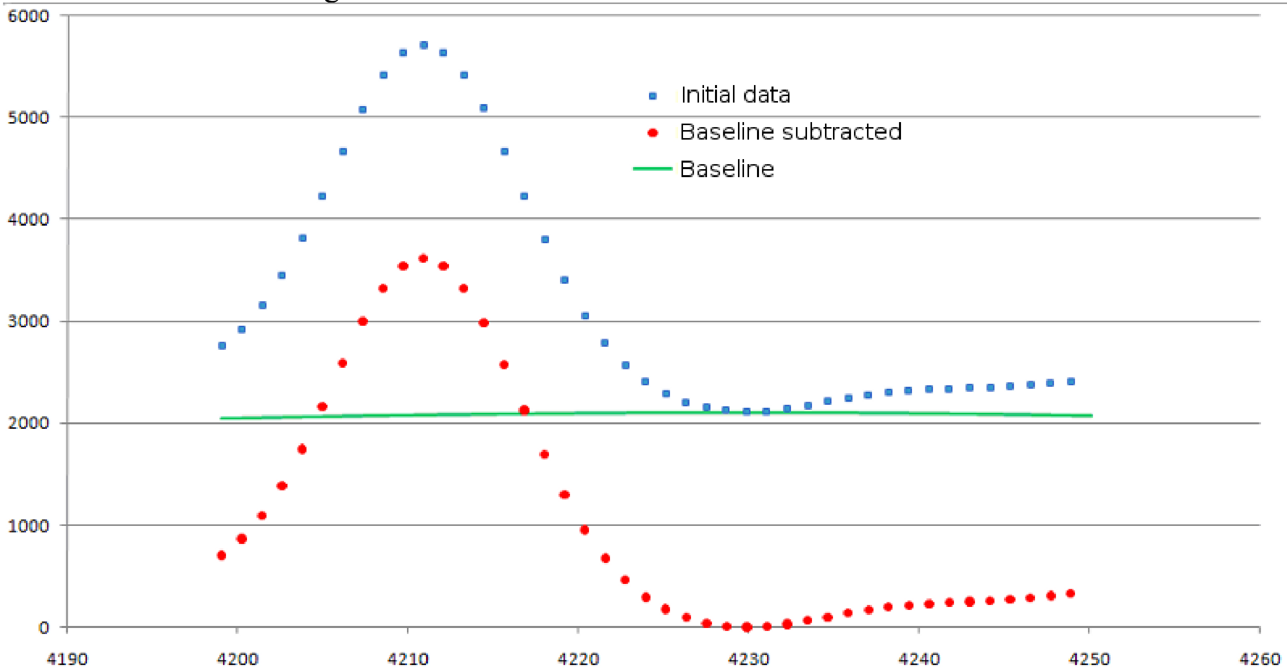


Figure 3. Result of background signal subtraction. Original data are shown as blue squares, modified ones - as red circles. Baseline is shown in green line.

Step 4. Normalization.

Normalization allows to bring peaks intensity values to a common scale, and thus it becomes possible to compare data from different spectra. The only parameter for current procedure is '*NormalizationConstant*' (default value is 10000). Example is shown in fig. 4.

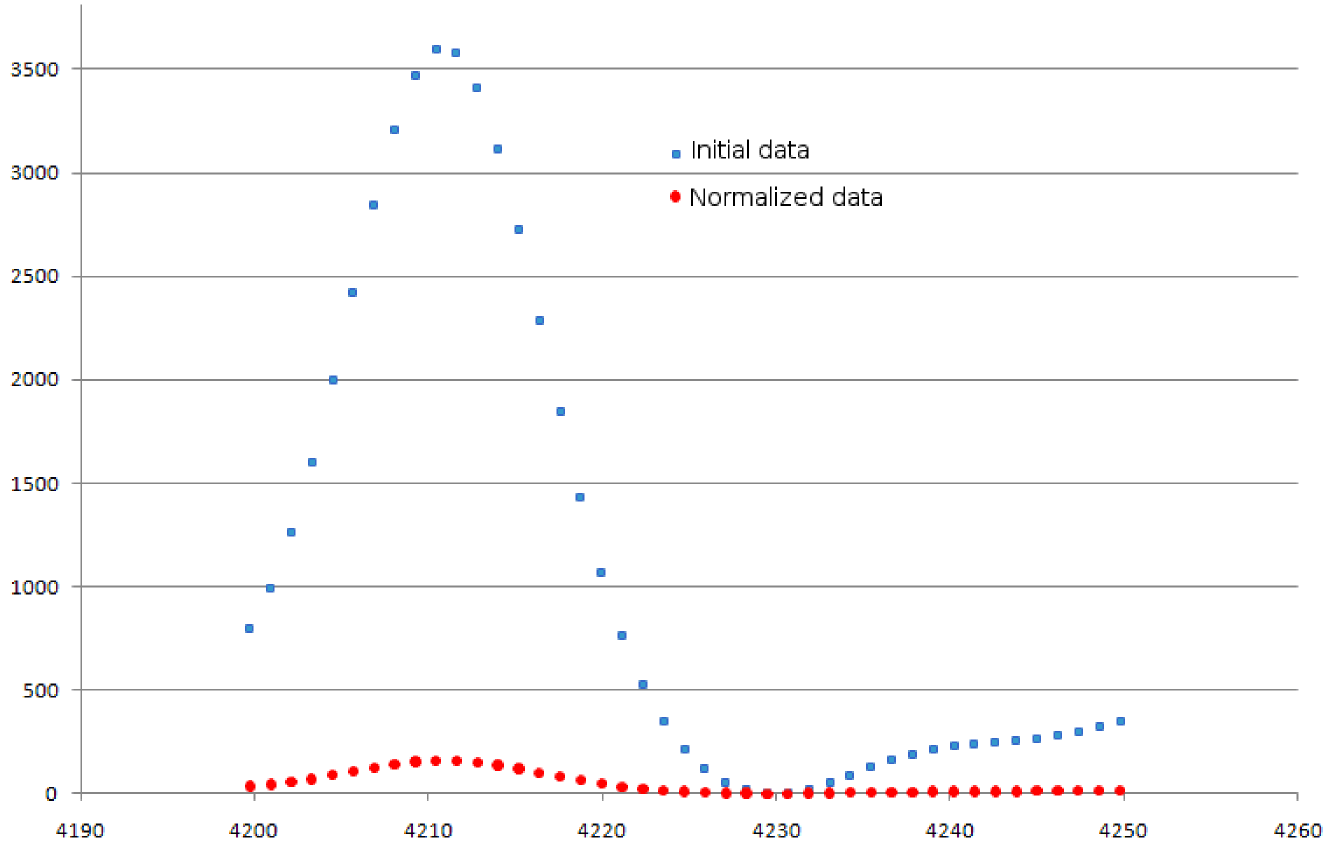


Figure 4. Result of normalization procedure. Original data are shown as blue squares, modified ones - as red circles.

Step 5. Peaks identification.

The current step of analysis lies in searching for peaks in spectrum with high signal-noise relation. Peaks, in themselves, are identified as points of local spectrum maximum. The '*SNRMin*' parameter specifies the minimally allowed value for signal-noise relation (default value is 3). This relation is considered in spectrum window w, size of which can be specified by the '*SNRWindowSize*' parameter (default value is 250). Thus, the value for signal-noise relation is calculated as:

$$SNR = I(m_i)/[\frac{1}{w}\Sigma_{i-w/2}^{i+w/2} I(m_i)]$$

Program identify peaks with*SNR*>'*SNRMin*' and intensity not less than '*MinIntensity*' (default value is 2). The result is shown in fig. 5.

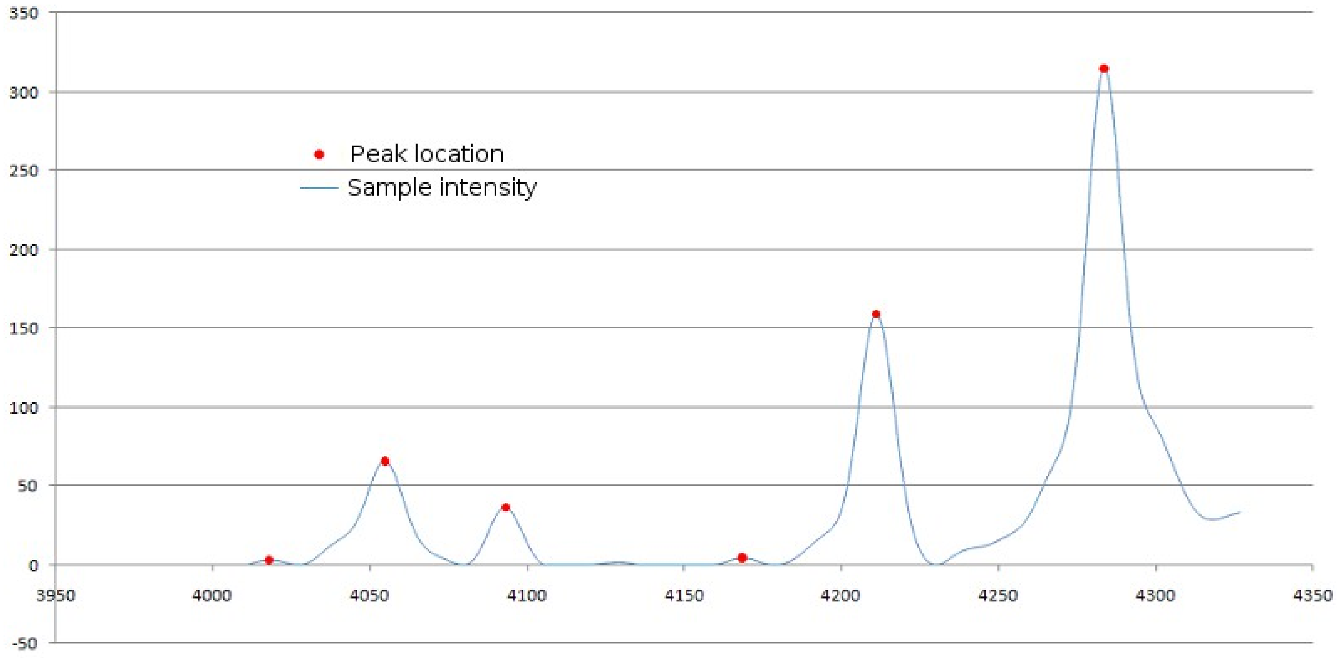


Figure 5. Identification of peaks in mass spectrum. Peaks locations over signal curve are marked with red dots.

Step 6. Peaks alignment.

On analyzing several spectra the question if there are common peaks for these spectra easily arises. To solve this question it is necessary to compare peaks locations and intensity for spectra of interest. It is mandatory that for all spectra to be compared the steps 1 to 5 are to be completed with the same parameters. For further analysis, the peaks with signal-noise ratio not lower than that specified by '*SNRMin*' parameter (default value is 3) will be selected. Once it is done, the peaks from different spectra are being grouped. Peak can be placed in the specific group if valuedm=$|m_i$-$mHi_{group}|/mHi_{group}$, where $mHi_{group}$is the maximal mass value for peaks in current group, does not exceed that specified by the '*MassSeparation*' parameter (default value is 0.0015). If multiple peaks groups meet this criterion, the group with minimal dm value is to be selected. Example for two spectra is shown in fig. 6.

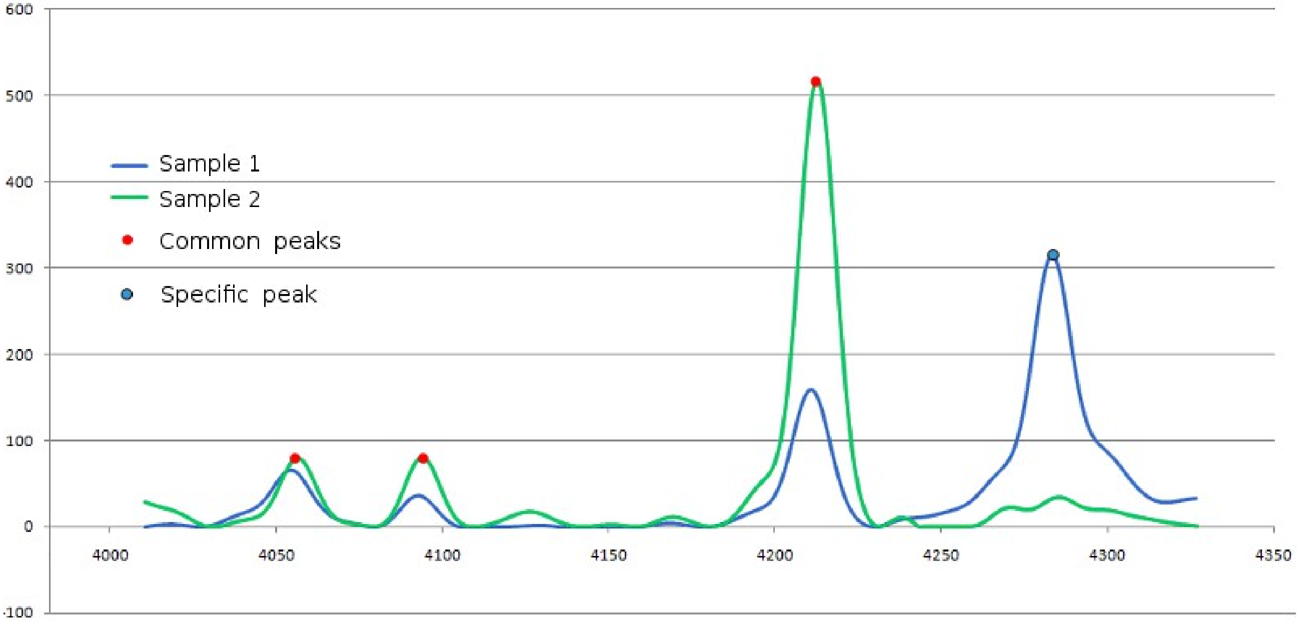


Figure 6. Alignment of peaks from two mass spectra. Spectra are represented by blue and green curves, common peaks are marked with red dots, sample 1 specific peak is marked with dot of blue color.

Input: m/z - Intensity data

Output: List of peak groups identified with the following information for each group: GroupIndex (Index of peak group), PeakID (peak ID), HighMass (highest value if*m/z*ration in peak group), MeanMass (mean value of*m/z*ration in the peak group), MinMass (minimal value of*m/z*ration in the peak group), MaxMass (maximal value of*m/z*ration in the peak group), NumPeaks (number of peaks in the peak group), MaxIntensity (maximal peak intensity in the peak group).

Parameter(s):

Binning percent- This parameter specify the fraction of data in percent that will remain after resampling. The default value is 25.

SmoothWindowSize- This parameter determine window size for smoothing operation. The default value is 3.

SmoothReps- This parameter specify the number of smoothing operation repeats. The default value is 3.

Baseline parameter- This parameter specify the minimal mass difference, over which the two neighboring peaks 1 and 2 are to be distinguished for baseline determination. The default value is 0.005.

NormalizationConstant- This parameter specify the normalization constant. The default value is 10000.

SNRWindowSize- This parameter specify window size to determine signal-to-noise ratio. The default value is 250.

SNRMin- This parameter specify minimal signal-to-noise ratio for peak detection. The default value is 3.

MinIntensity- This parameter specify minimal intensity for peak detection. The default value is 2.

MassSeparation- This parameter specify minimal mass separation for peaks from the same group. The default value is 0.0015.

File format type- This parameter specify file format. SSV-space separated values, CSV - comma separated values, TSV - tab separated values.

Data format.

Mass spectra data represent the sets of following pairs of values: mass to charge relation (m/z, further, for more convenience, it will be referred to as m, mass) and corresponding signal intensity (I). On a spectrum plot, the mass corresponds to X coordinate, and signal intensity- to Y one. A typical spectrum consists of several thousand of such value pairs (points). Data are represented as text files, where for each pair (m_i,I_i) of mass-intensity values the string is assigned, and data in this string are separated by special separator symbol. The SMS package allows several separators types: space (SSV, space separated values, file format), comma (CSV, comma separated values, file format) and tabulation (TSV, tab-separated values, file format). In files with data, the string with comments are allowed; during the file reading these strings are to be skipped. The commentary strings should begin with "#" symbol at the first position. In the figure 2 the example of file with data in CSV format is shown.

```
#M/Z,Intensity
-7.8602611e-005,4.1126194
2.1773576e-007,4.0764203
9.6021472e-005,4.0040221
0.00036601382,4.1186526
0.00081019477,4.0040221
```

```
0.0014285643,3.9617898
….
19742.941,4.077895
19745.564,4.0772248
19748.187,4.0772248
```

Figure 7. Example file with mass spectra data in CSV format.

Parameters:

Input	
Input Data Folder	The name of the directory with input data files.
Data Files Set	Text file contain list of files with different spectra (one per line).
Input Files Format	This parameter specify the format of input files: Space separated values, Comma separated values, Tab separated values.
Output	
Result	Name of output file
Options	
Binning percent	This parameter specify the fraction of data in percent that will remain after resampling.
Smoothing repeats	This parameter specify the number of smoothing operation repeats.
Smoothing window size	This parameter determine window size for smoothing operation.
Baseline parameter	This parameter specify the minimal relative mass difference, over which the two neighboring peaks 1 and 2 are to be distinguished for baseline determination.
Normalization constant	This parameter specify the normalization constant.
SNR window size	This parameter specify window size to determine signal-to-noise ratio.
SNR minimum	This parameter specify minimal signal-to-noise ratio for peak detection.
Minimal peak intensity	This parameter specify minimal intensity for peak intensity.
Minimal mass separation	This parameter specify minimal mass separation for peaks assignment in the calibration data.

MSPeakFind

Proteomics-MSPeakFind- Softberry Mass Spectra (SMS) processing tools. Peaks identification.
On a single spectrum processing, this program performs the following operations:

(1) Data resampling;

(2) Data smoothing;

(3) Detection of the baseline and its subtraction from intensity;

(4) Normalization;

(5) Peaks identification.

Step 1. Data resampling.

The first step in mass spectra processing is data resampling. It allows to discriminate the excessive data and to bring the$m_i$values to common scale. As a result, different spectra will have the same$m$value counts, and, thus, will be comparable. Reduction in number of spectrum points allows to lower the noise and to eliminate excessive data, but, at the same time, to keep the spectrum shape. The common data scale after conversion is located between the minimal and maximal$m$values of spectrum. The number of data that will be resampled from original set is determined by the '*Binning percent*' parameter, that represents the percentage of spectrum points remained after conversion (default value is 25). Example of data resampling is shown in figure 1.

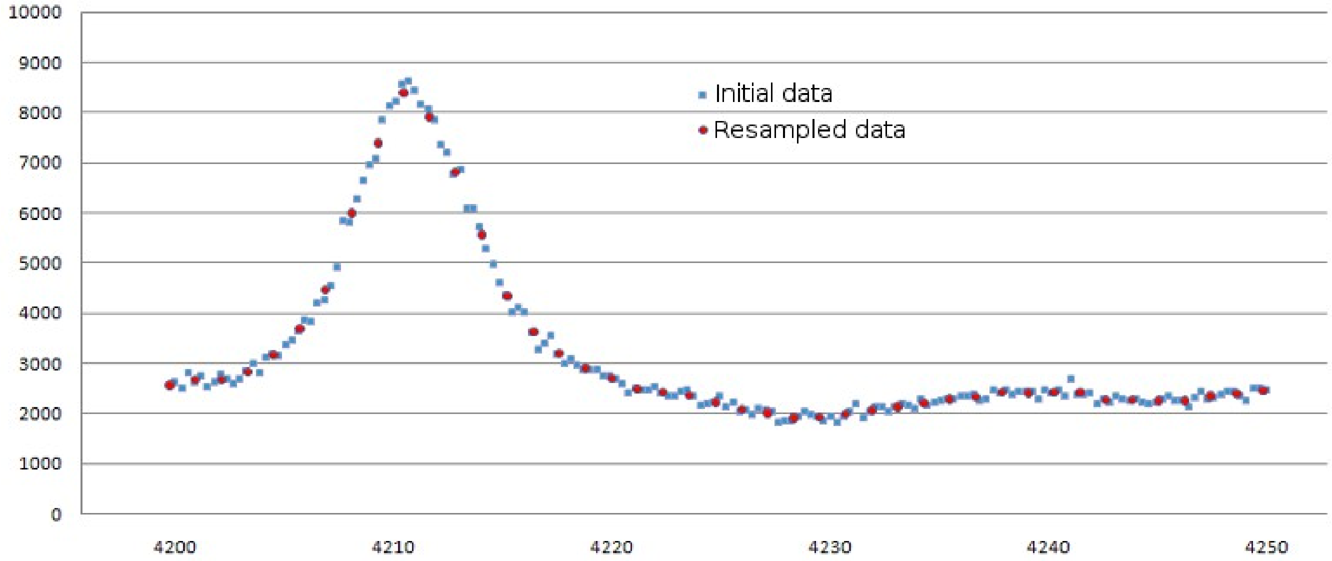


Figure 1. Result of data resampling for small spectrum interval. Original data are shown as blue squares, resampled ones - as red circles. The '*Binning percent*' for this case was set to 25.

Step 2. Smoothing.

Data smoothing procedure is intended for data noise elimination. During the smoothing, the values of intensity for each$m_i$point are being averaged by several neighboring points. The number of such points is determined by the '*SmoothWindowSize*' parameter (default value is 3). The smoothing procedure can be repeated for several times; the number of iterations is determined by the '*SmoothReps*' parameter (default value is 3). Example of data smoothing is shown in the figure 2.

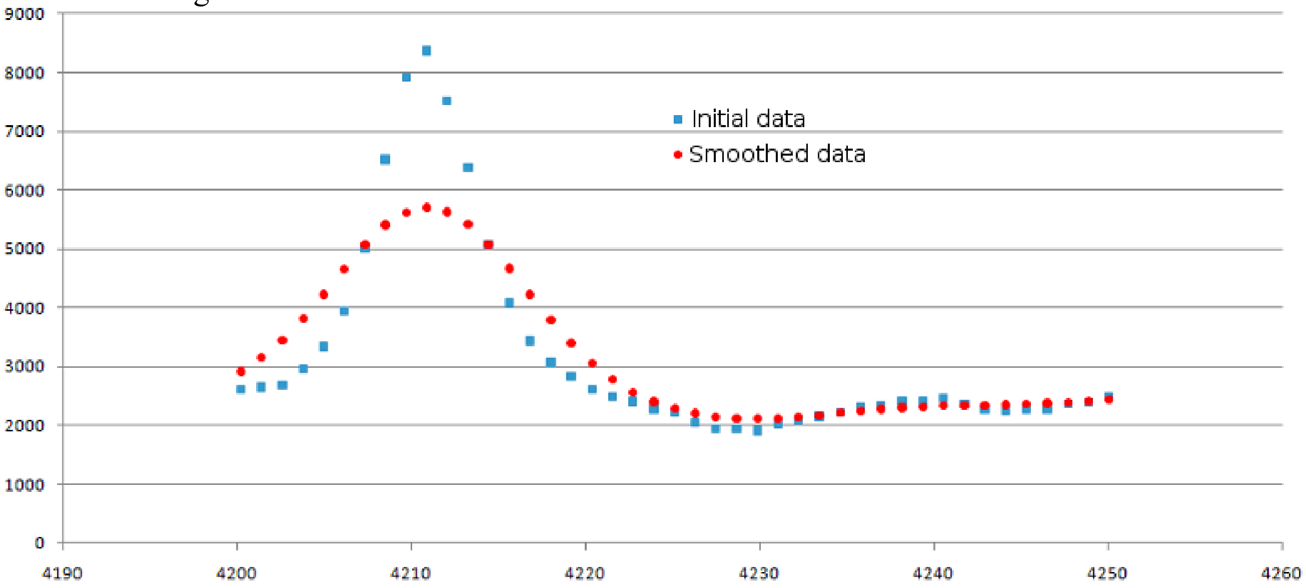


Figure 2. Result of data smoothing. Original data are shown as blue squares, smoothed ones - as red circles. The*SmoothWindowSize*was set to 3 and*SmoothReps*was set to 3.

Step 3. Baseline detection and subtraction.

This step of data processing is applied for elimination of the systematic artifacts that occur due to matrix and chemicals used in the experiments or as a result of detector overload. It results in background noise that may occur to be significant for some*m*values. The initial step in background noise removal is identification of peaks (local signal maxima that are located far enough from each other). The distance between peaks is determined by the '*Baseline parameter*' value (default= 0.005). This parameter defines the minimal*m*distance, over which the two neighboring peaks 1 and 2 are to be located in the way, when:

$|m_1\text{-}m_2|/m_1$> '*Baseline parameter*'.

After peaks identification, algorithm detects the points with signal minima located in intervals between peaks. These are the base points for calculation of background noise line. Over base points the baseline for all spectrum points is built by interpolation. In case when in some spectrum parts the value of base signal exceeds the original one, the new base points selection from neighboring ones occurs.

The values of base signal intensity are subtracted from the original one. At that, if value of original signal has occurred below zero, it is equated to zero. The result of background subtraction is shown in figure 3.

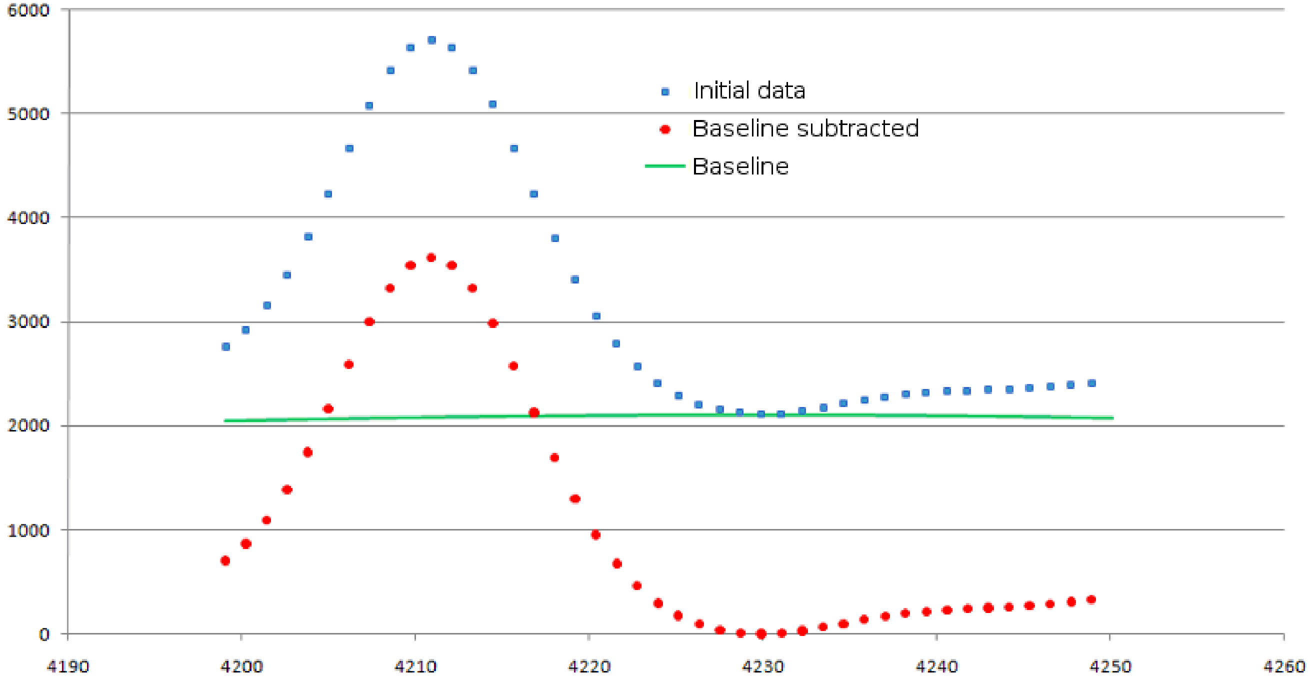


Figure 3. Result of background signal subtraction. Original data are shown as blue squares, modified ones - as red circles. Baseline is shown in green line.

Step 4. Normalization.

Normalization allows to bring peaks intensity values to a common scale, and thus it becomes possible to compare data from different spectra. The only parameter for current procedure is '*NormalizationConstant*' (default value is 10000). Example is shown in fig. 4.

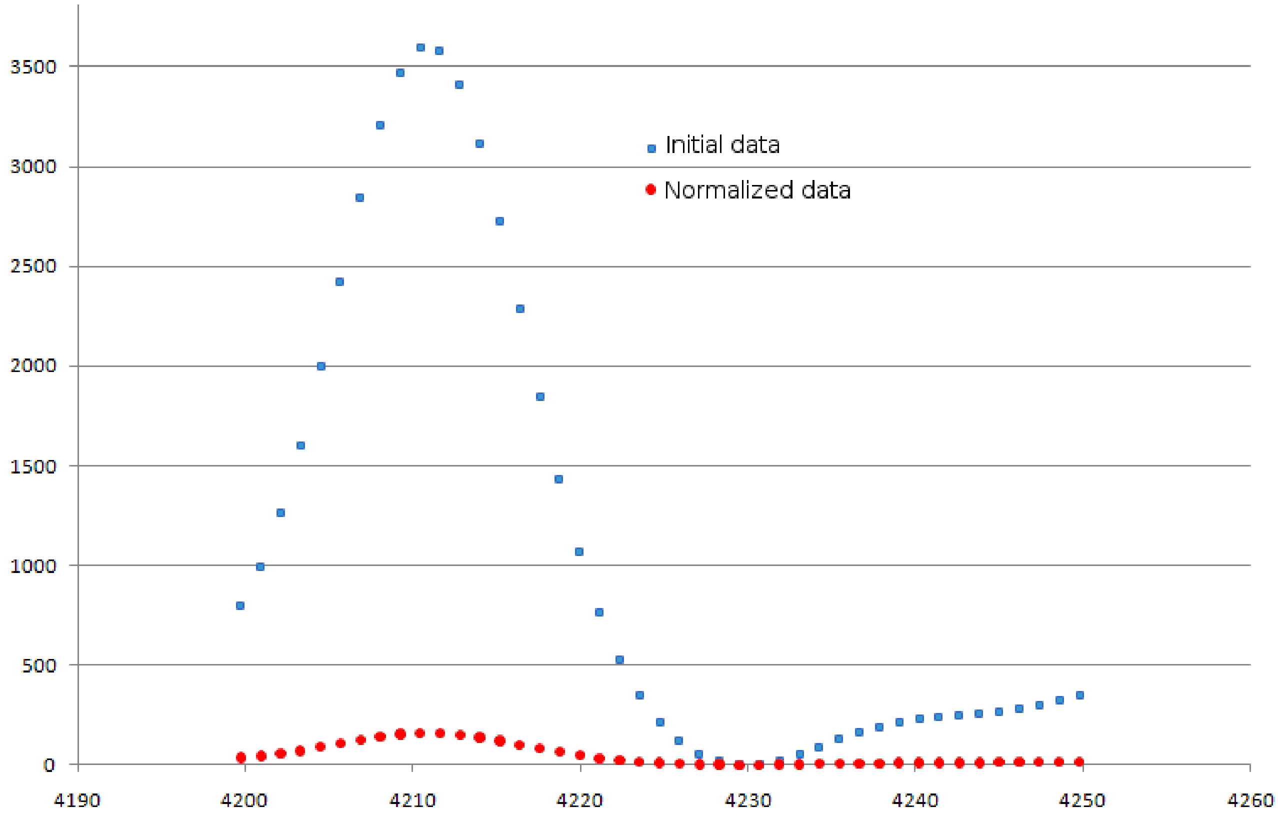


Figure 4. Result of normalization procedure. Original data are shown as blue squares, modified ones - as red circles.

Step 5. Peaks identification.

The current step of analysis lies in searching for peaks in spectrum with high signal-noise relation. Peaks, in themselves, are identified as points of local spectrum maximum. The '*SNRMin*' parameter specifies the minimally allowed value for signal-noise relation (default value is 3). This relation is considered in spectrum window w, size of which can be specified by the '*SNRWindowSize*' parameter (default value is 250). Thus, the value for signal-noise relation is calculated as:

$$SNR = I(m_i)/[\frac{1}{w}\sum_{i-w/2}^{i+w/2} I(m_i)]$$

Program identify peaks with*SNR*>'*SNRMin*' and intensity not less than '*MinIntensity*' (default value is 2). The result is shown in fig. 5.

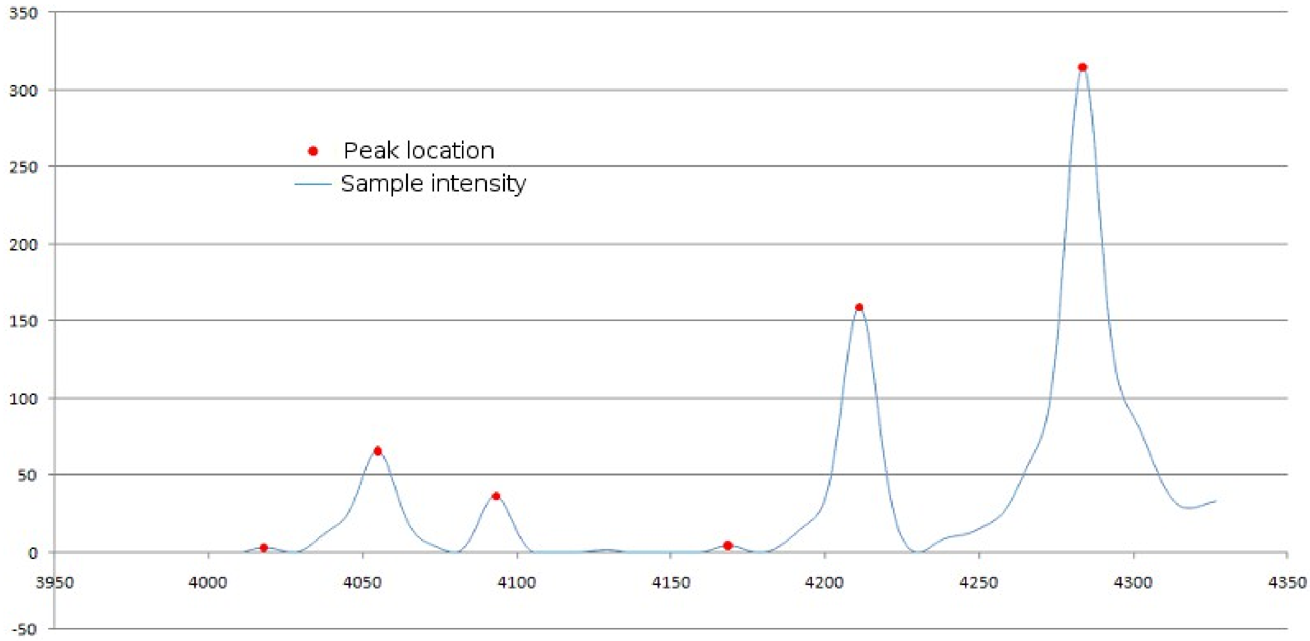


Figure 5. Identification of peaks in mass spectrum. Peaks locations over signal curve are marked with red dots.

Input: m/z - Intensity data

Output: List of peaks with the following information for each peak:

PeakIndex (Peak ID), SampleIndex (Index of sample, for this program is always 0), PointNum (number of point corresponding to peak location in the sample), MZRatio (m/z value), SNRRatio (signal to noise ratio), Intensity (signal intensity).

Parameter(s):

Binning percent- This parameter specify the fraction of data in percent that will remain after resampling. The default value is 25.

SmoothWindowSize- This parameter determine window size for smoothing operation. The default value is 3.

SmoothReps- This parameter specify the number of smoothing operation repeats. The default value is 3.

Baseline parameter- This parameter specify the minimal mass difference, over which the two neighboring peaks 1 and 2 are to be distinguished for baseline determination. The default value is 0.005.

NormalizationConstant- This parameter specify the normalization constant. The default value is 10000.

SNRWindowSize- This parameter specify window size to determine signal-to-noise ratio. The default value is 250.

SNRMin- This parameter specify minimal signal-to-noise ratio for peak detection. The default value is 3.

MinIntensity- This parameter specify minimal intensity for peak detection. The default value is 2.

File format type- This parameter specify file format. SSV-space separated values, CSV - comma separated values, TSV - tab separated values.

Data format.

Mass spectra data represent the sets of following pairs of values: mass to charge relation (m/z, further, for more convenience, it will be referred to as m, mass) and corresponding signal intensity (I). On a spectrum plot, the mass corresponds to X coordinate, and signal intensity- to Y one. A typical spectrum consists of several thousand of such value pairs (points). Data are represented as text files, where for each pair (m_i,I_i) of mass-intensity values the string is assigned, and data in this string are separated by special separator symbol. The SMS package allows several separators types: space (SSV, space separated values, file format), comma (CSV, comma separated values, file format) and tabulation (TSV, tab-separated values, file format). In files with data, the string with comments are allowed; during the file reading these strings are to be skipped. The commentary strings should begin with "#" symbol at the first position. In the figure 2 the example of file with data in CSV format is shown.

```
#M/Z,Intensity
-7.8602611e-005,4.1126194
2.1773576e-007,4.0764203
9.6021472e-005,4.0040221
0.00036601382,4.1186526
0.00081019477,4.0040221
0.0014285643,3.9617898
….
19742.941,4.077895
19745.564,4.0772248
19748.187,4.0772248
```

Figure 7. Example file with mass spectra data in CSV format.

Parameters:

Input	
Input data file	File with input data.
Input Files Format	This parameter specify the format of input files: Space separated values, Comma separated values, Tab separated values.
Output	
Result	Name of output file
Options	
Binning percent	This parameter specify the fraction of data in percent that will remain after resampling.
Smoothing repeats	This parameter specify the number of smoothing operation repeats.
Smoothing window size	This parameter determine window size for smoothing operation.
Baseline parameter	This parameter specify the minimal relative mass difference, over which the two neighboring peaks 1 and 2 are to be distinguished for baseline determination.
Normalization constant	This parameter specify the normalization constant.
Normalization constant	This parameter specify the normalization constant.
SNR window size	This parameter specify window size to determine signal-to-noise ratio.
SNR minimum	This parameter specify minimal signal-to-noise ratio for peak detection.
Minimal peak intensity	This parameter specify minimal intensity for peak intensity.

MSPredictLDA

MSPredictLDAprogram performs classification of patient for cancer/normal case using the mass-spectrum data and CA125 marker level.

Parameters:

Input	
File with input data	Text file should contain two columns separated by separating character (see input format type parameter). First column - m/z ratio (mass), second - Intensity.
Input Files Format	This parameter specify the format of input files: Space separated values, Comma separated values, Tab separated values.
LDF File	This file contain LDF calculation parameters obtained as result of MSCalcParamLDA module.
Output	
Result	Name of output file
Options	
Binning percent	This parameter specify the fraction of data in percent that will remain after resampling.
Smoothing repeats	This parameter specify the number of smoothing operation repeats.
Smoothing window size	This parameter determine window size for smoothing operation.
Baseline parameter	This parameter specify the minimal relative mass difference, over which the two neighboring peaks 1 and 2 are to be distinguished for baseline determination.
Normalization constant	This parameter specify the normalization constant.
CA125 Level	This parameter define the level of the CA125 marker needed to classify samples for cancer/non-cancer types.

MSPreprocess

MSPreprocess - program performs preprocessing steps for the mass-spectrum data.

Parameters:

Input	
Input data file	File with input data.
Input Files Format	This parameter specify the format of input files: Space separated values, Comma separated values, Tab separated values.
Output	
Result	Name of output file
Options	
Binning percent	This parameter specify the fraction of data in percent that will remain after resampling.
Smoothing repeats	This parameter specify the number of smoothing operation repeats.
Smoothing window size	This parameter determine window size for smoothing operation.
Baseline parameter	This parameter specify the minimal relative mass difference, over which the two neighboring peaks 1 and 2 are to be distinguished for baseline determination.
Normalization constant	This parameter specify the normalization constant.

MSResampling

MSResampling - this program performs resampling of the mass-spectrum data

The first step in mass spectra processing is data resampling. It allows to discriminate the excessive data and to bring the$m_i$values to common scale. As a result, different spectra will have the same$m$value counts, and, thus, will be comparable. Reduction in number of spectrum points allows to lower the noise and to eliminate excessive data, but, at the same time, to keep the spectrum shape. The common data scale after conversion is located between the minimal and maximal$m$values of spectrum. The number of data that will be resampled from original set is determined by the '*Binning percent*' parameter, that represents the percentage of spectrum points remained after conversion (default value is 25). Example of data resampling is shown in figure 1.

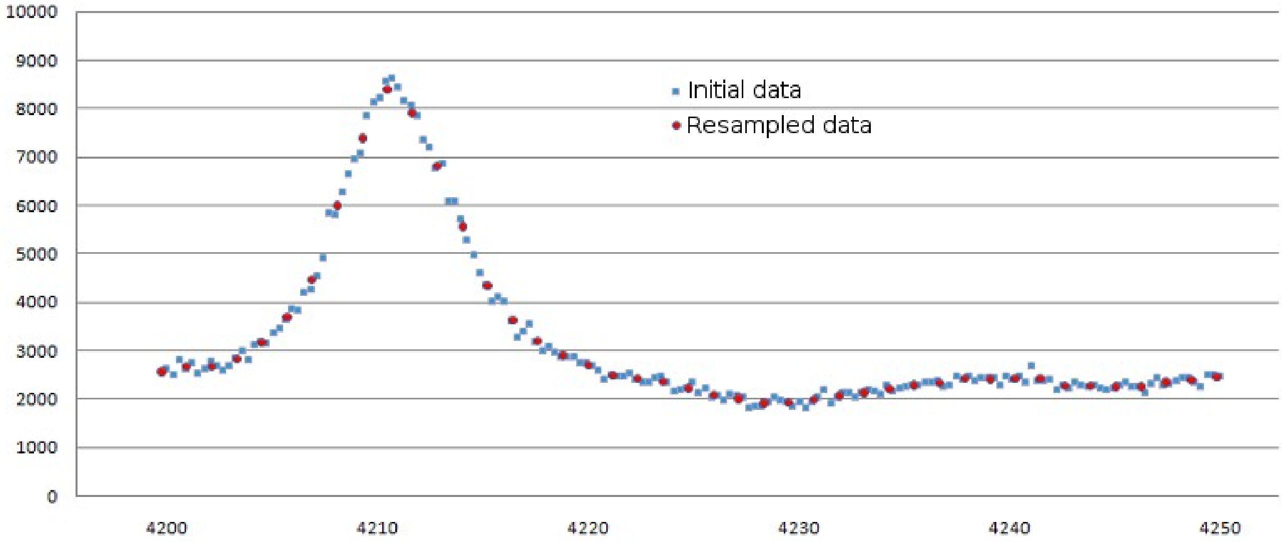


Figure 1. Result of data resampling for small spectrum interval. Original data are shown as blue squares, resampled ones - as red circles. The '*Binning percent*' for this case was set to 25.

Input: m/z - Intensity data

Output: Resampled m/z - Intensity data in the same format as input data.

Parameter(s):

Binning percent- This parameter specify the fraction of data in percent that will remain after resampling. The default value is 25.

Data format.

Mass spectra data represent the sets of following pairs of values: mass to charge relation (m/z, further, for more convenience, it will be referred to as m, mass) and corresponding signal intensity (I). On a spectrum plot, the mass corresponds to X coordinate, and signal intensity- to Y one. A typical spectrum consists of several thousand of such value pairs (points). Data are represented as text files, where for each pair (m_i,I_i) of mass-intensity values the string is assigned, and data in this string are separated by special separator symbol. The SMS package allows several separators types: space (SSV, space separated values, file format), comma (CSV, comma separated values, file format) and tabulation (TSV, tab-separated values, file format). In files with data, the string with comments are allowed; during the file reading these strings are to be skipped. The commentary strings should begin with "#" symbol at the first position. In the figure 2 the example of file with data in CSV format is shown.

```
#M/Z,Intensity
-7.8602611e-005,4.1126194
2.1773576e-007,4.0764203
9.6021472e-005,4.0040221
```

```
0.00036601382,4.1186526
0.00081019477,4.0040221
0.0014285643,3.9617898
….
19742.941,4.077895
19745.564,4.0772248
19748.187,4.0772248
```

Figure 2. Example file with mass spectra data in CSV format.

Parameters:

Input	
Input data file	File with input data.
Input Files Format	This parameter specify the format of input files: Space separated values, Comma separated values, Tab separated values.
Output	
Result	Name of output file
Options	
Binning percent	This parameter specify the fraction of data in percent that will remain after resampling.

MSSmoothing

Proteomics-MSSmoothing- Softberry Mass Spectra (SMS) processing tools. Data smoothing. Data smoothing procedure is intended for data noise elimination. During the smoothing, the values of intensity for each mi point are being averaged by several neighboring points. The number of such points is determined by the '*SmoothWindowSize*' parameter (default value is 3). The smoothing procedure can be repeated for several times; the number of iterations is determined by the '*SmoothReps*' parameter (default value is 3). Example of data smoothing is shown in the figure 1.

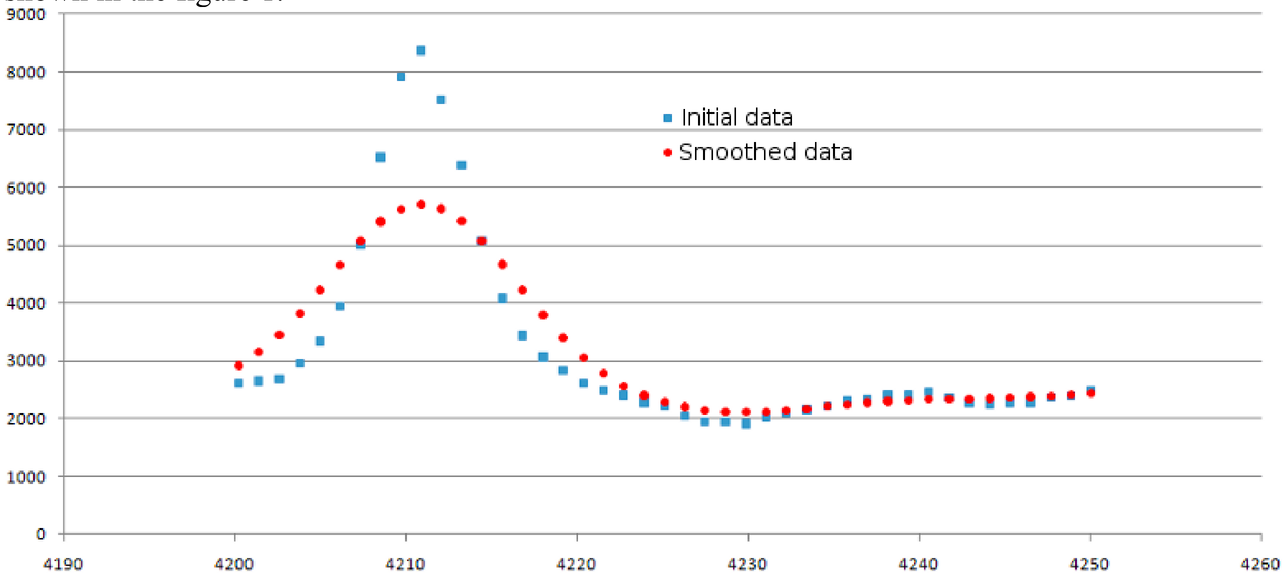


Figure 1. Result of data smoothing. Original data are shown as blue squares, smoothed ones - as red circles. The*SmoothWindowSize*was set to 3 and*SmoothReps*was set to 3.

Input: m/z - Intensity data

Output: Smoothed m/z - Intensity data in the same format as input data.

Parameter(s):

SmoothWindowSize- This parameter determine window size for smoothing operation. The default value is 3.

SmoothReps- This parameter specify the number of smoothing operation repeats. The default value is 3.

File format type- This parameter specify file format. SSV-space separated values, CSV - comma separated values, TSV - tab separated values.

Data format.

Mass spectra data represent the sets of following pairs of values: mass to charge relation (m/z, further, for more convenience, it will be referred to as m, mass) and corresponding signal intensity (I). On a spectrum plot, the mass corresponds to X coordinate, and signal intensity- to Y one. A typical spectrum consists of several thousand of such value pairs (points). Data are represented as text files, where for each pair (m_i,I_i) of mass-intensity values the string is assigned, and data in this string are separated by special separator symbol. The SMS package allows several separators types: space (SSV, space separated values, file format), comma (CSV, comma separated values, file format) and tabulation (TSV, tab-separated values, file format). In files with data, the string with comments are allowed; during the file reading these strings are to be skipped. The commentary strings should begin with "#" symbol at the first position. In the figure 2 the example of file with data in CSV format is shown.

```
#M/Z,Intensity
-7.8602611e-005,4.1126194
2.1773576e-007,4.0764203
9.6021472e-005,4.0040221
0.00036601382,4.1186526
0.00081019477,4.0040221
0.0014285643,3.9617898
….
19742.941,4.077895
19745.564,4.0772248
19748.187,4.0772248
```

Figure 2. Example file with mass spectra data in CSV format.

Parameters:

Input	
Input data file	File with input data.
Input Files Format	This parameter specify the format of input files: Space separated values, Comma separated values, Tab separated values.
Output	
Result	Name of output file
Options	
Smoothing repeats	This parameter specify the number of smoothing operation repeats.
Smoothing window size	This parameter determine window size for smoothing operation.

RNA Structure

BestPal-E

Calculates the best palindrome for given rna sequence, and also a set suboptimal palindromes (sorted by energy)

Method description:

First the complementary matrix is built, and all helixes are detected. Then they are sorted by their stability. Then starting each structure with one of most stable helixes from sorted list (each time different from others), the program upgrades them with compatible helixes until adding new helix gives no stability growth or when there are no more compatible helixes. Best N structures are written to user-defined file.

Output example:

```
==== structure 1 ====
Start   End     Energy
   24    996     -173.6
Helices: 29
   24 -   25   AC
  996 -  995   UG

   31 -   33   UCA
  991 -  989   AGU

   36 -   38   UCA
  984 -  982   AGU

   42 -   43   GA
  978 -  977   CU

   45 -   52   UGAUCGAU
  975 -  968   GCUAGCUA

   55 -   65   CUAGCUAGCUG
  962 -  952   GAUCGAUCGAU

   68 -   69   AC
  948 -  947   UG

   74 -   78   UGAUC
  943 -  939   GCUAG

  176 -  178   GUG
  937 -  935   UAC

  185 -  189   GCUAC
  928 -  924   CGAUG

  214 -  225   GUCGUACGUAGC
  918 -  907   UAGCAUGCAUCG

  503 -  513   AUCGUACGUAC
  906 -  896   UAGCAUGCAUG

  526 -  528   CUC
  891 -  889   GGG

  531 -  538   UACGUACG
  884 -  877   AUGCAUGC

  539 -  543   UACGC
  847 -  843   GUGUG

  550 -  561   GCUACGUACGUG
  835 -  824   CGAUGCAUGCAU

  562 -  565   ACUG
  806 -  803   UGAU

  569 -  571   GCA
  798 -  796   CGU

  582 -  587   GUGCAU
  793 -  788   UACGUA

  593 -  596   CGAU
  779 -  776   GCUA

  598 -  602   ACUGU
  770 -  766   UGAUG

  608 -  620   UAGCAUGCAUCGA
  760 -  748   AUCGUACGUAGCU

  621 -  622   GC
  741 -  740   CG

  627 -  629   GGC
  734 -  732   UCG

  631 -  636   GUCAGC
  727 -  722   UAGUCG

  639 -  641   GGU
  716 -  714   UCG

  642 -  648   GCUACGU
  705 -  699   CGAUGCA

  660 -  665   UGAUCG
  697 -  692   GCUAGU

  670 -  672   UAG
  686 -  684   AUC

==== structure 2 ====
Start   End    Energy
    3    998    -172.1
Helices: 24
    3 -    8   GUACUA
  998 -  993   CAUGGU

   12 -   14   GUG
  988 -  986   CAU

   23 -   24   CA
  983 -  982   GU

   28 -   32   UGAUC
  979 -  975   GCUAG

   45 -   52   UGAUCGAU
  971 -  964   GCUAGCUA

   55 -   65   CUAGCUAGCUG
```

```
 958 -  948   GAUCGAUCGAU

  74 -   78   UGAUC
 943 -  939   GCUAG

 178 -  180   GUG
 937 -  935   UAC

 185 -  189   GCUAC
 928 -  924   CGAUG

 214 -  225   GUCGUACGUAGC
 918 -  907   UAGCAUGCAUCG

 503 -  513   AUCGUACGUAC
 906 -  896   UAGCAUGCAUG

 526 -  528   CUC
 891 -  889   GGG

 531 -  538   UACGUACG
 884 -  877   AUGCAUGC

 539 -  543   UACGC
 847 -  843   GUGUG

 550 -  561   GCUACGUACGUG
 835 -  824   CGAUGCAUGCAU

 567 -  570   CUGC
 816 -  813   GAUG

 578 -  583   ACUAGU
 806 -  801   UGAUCG

 607 -  620   GUAGCAUGCAUCGA
 798 -  785   CGUCGUACGUAGCU

 626 -  628   CGG
 783 -  781   GCU

 631 -  636   GUCAGC
 777 -  772   UAGUCG

 641 -  643   UGC
 771 -  769   AUG

 698 -  709   UACGUAGCUAGU
 768 -  757   AUGCAUCGAUCG

 714 -  715   GC
 754 -  753   CG

 720 -  725   UAGCUG
 743 -  738   AUCGAU
 ..........
```

Parameters:

<table>
<tr><th colspan="2">Input</th></tr>
<tr><td><b>Sequence</b></td><td>File with RNA sequence.</td></tr>
<tr><th colspan="2">Output</th></tr>
<tr><td><b>Result</b></td><td>Output file.</td></tr>
<tr><th colspan="2">Options</th></tr>
</table>

Number of structures	Number of secondary structures for output.

BestPal-H

Calculates best palindrome for given rna sequence with restrictions.
In this version two types of restriction can be specified:

1) minimal helix length allowed
2) maximal secondary structure length allowed

Method description:

Dynamic programming method without "brahching" of structures with filters using specified restrictions.

Output example:

```
Search for most stable hairpin (imperfect helices included)

FoldRNA Vienna format:
Length: 754 Energy: -7.8 3% in Helices

        10        20        30        40        50        60
UGCGGCGGAGACCGUGGUUUAGUGGGCCAAGGGUUCUACGAGUCGGAACACGUGUUAUCU
..((((...((((.((((((...))))))..))))......))))...............
        70        80        90       100       110       120
CUUGCGAAGAGUUUAAGGGUCCUGAGGGUGCGGAGUUGUGUUAUCAACCGAACACAGAAG
............................................................
       130       140       150       160       170       180
AAUCCCAAAUGAUGAAGCUGAGUCUCAUCAAAGUCGUUAAUGGCUGUCGUCUAGGAAAAA
............................................................
       190       200       210       220       230       240
UACAAAACCUGGGCAAAGCAGGGGACUGCACGGUGGACAUUCCGGGCUGUCUUCUCUACA
............................................................
       250       260       270       280       290       300
CCAGGACUGGCUCUGCCCCACACCUGACACAUCAGACGCUGCGUAACAUCCACGGGGUCC
............................................................
       310       320       330       340       350       360
CAGGCAUAGCCCAGCUCACACUCUCAUCCCUAGCAGAACAUCAUGAAGUCUUGGCAGAAU
............................................................
       370       380       390       400       410       420
AUAAGAAAGGAGUUGGAAGCUUUAUAGGCAUGCCGGAAUCACUCUUCUAUUGUUCCCUGC
............................................................
       430       440       450       460       470       480
ACGAUCCAGUCACCCCCGGCCCAGCUGGUUAUGUAACAAGUAAGGUCCUCCAGAAAAGUG
............................................................
       490       500       510       520       530       540
UGAUCAUUGGAGUGAUUGAGGGUGGAGAUGUGAUGGAAGAGAGGUUGAGGUCAGCACGAG
............................................................
       550       560       570       580       590       600
AGACAGCCAAGCGACCCGUCGGGGGCUUCCUGCUGGACGGCUUUCAAGGGGAUCCAGCAG
............................................................
       610       620       630       640       650       660
UCACAGAAACCAGACUGCACUUGCUGUCAUCAGUCACUGCAGAGCUGCCAGAGGACAAAC
............................................................
       670       680       690       700       710       720
CAAGGCUCAUCUGCGGUGUCAGCCGGCCAGACGAAGUGCUAGAGUGCAUCGAAAGGGGAG
............................................................
       730       740       750       760
UGGACUUGUUUGAGAGUUUUUUCCCAUAUCAAGU
..................................


Length = 754
==== structure 1 ====
Start   End     Energy
```

```
    3     45     -7.8
Helices: 3
    3 -    6   CGGC
   42 -   45   GCUG

   10 -   13   GACC
   32 -   35   UUGG

   15 -   20   UGGUUU
   24 -   29   ACCGGG



FoldRNA GCG format:
Length: 754 Energy: -7.8

    1 U       0    2    0    1
    2 G       1    3    0    2
    3 C       2    4   45    3
    4 G       3    5   44    4
    5 G       4    6   43    5
    6 C       5    7   42    6
    7 G       6    8    0    7
    8 G       7    9    0    8
    9 A       8   10    0    9
   10 G       9   11   35   10
   11 A      10   12   34   11
   12 C      11   13   33   12
   13 C      12   14   32   13
   14 G      13   15    0   14
   15 U      14   16   29   15
   16 G      15   17   28   16
   17 G      16   18   27   17
   18 U      17   19   26   18
   19 U      18   20   25   19
   20 U      19   21   24   20
   21 A      20   22    0   21
   22 G      21   23    0   22
   23 U      22   24    0   23
   24 G      23   25   20   24
   25 G      24   26   19   25
   26 G      25   27   18   26
   27 C      26   28   17   27
   28 C      27   29   16   28
   29 A      28   30   15   29
   30 A      29   31    0   30
   31 G      30   32    0   31
   32 G      31   33   13   32
   33 G      32   34   12   33
   34 U      33   35   11   34
   35 U      34   36   10   35
   36 C      35   37    0   36
   37 U      36   38    0   37
   38 A      37   39    0   38
   39 C      38   40    0   39
   40 G      39   41    0   40
   41 A      40   42    0   41
   42 G      41   43    6   42
   43 U      42   44    5   43
   44 C      43   45    4   44
   45 G      44   46    3   45
   46 G      45   47    0   46
   47 A      46   48    0   47
   48 A      47   49    0   48
```

```
 49 C      48   50    0   49
 50 A      49   51    0   50
 51 C      50   52    0   51
 52 G      51   53    0   52
 53 U      52   54    0   53
 54 G      53   55    0   54
 55 U      54   56    0   55
 56 U      55   57    0   56
 57 A      56   58    0   57
 58 U      57   59    0   58
 59 C      58   60    0   59
 60 U      59   61    0   60
 61 C      60   62    0   61
 62 U      61   63    0   62
 63 U      62   64    0   63
 64 G      63   65    0   64
 65 C      64   66    0   65
 66 G      65   67    0   66
 67 A      66   68    0   67
 68 A      67   69    0   68
 69 G      68   70    0   69
 70 A      69   71    0   70
 71 G      70   72    0   71
 72 U      71   73    0   72
 73 U      72   74    0   73
 74 U      73   75    0   74
 75 A      74   76    0   75
 76 A      75   77    0   76
 77 G      76   78    0   77
 78 G      77   79    0   78
 79 G      78   80    0   79
 80 U      79   81    0   80
 81 C      80   82    0   81
 82 C      81   83    0   82
 83 U      82   84    0   83
 84 G      83   85    0   84
 85 A      84   86    0   85
 86 G      85   87    0   86
 87 G      86   88    0   87
 88 G      87   89    0   88
 89 U      88   90    0   89
 90 G      89   91    0   90
 91 C      90   92    0   91
 92 G      91   93    0   92
 93 G      92   94    0   93
 94 A      93   95    0   94
 95 G      94   96    0   95
 96 U      95   97    0   96
 97 U      96   98    0   97
 98 G      97   99    0   98
 99 U      98  100    0   99
100 G      99  101    0  100
101 U     100  102    0  101
102 U     101  103    0  102
103 A     102  104    0  103
104 U     103  105    0  104
105 C     104  106    0  105
106 A     105  107    0  106
107 A     106  108    0  107
108 C     107  109    0  108
109 C     108  110    0  109
110 G     109  111    0  110
111 A     110  112    0  111
112 A     111  113    0  112
```

```
  113 C     112  114    0  113
  114 A     113  115    0  114
  115 C     114  116    0  115
  116 A     115  117    0  116
  117 G     116  118    0  117
  118 A     117  119    0  118
  119 A     118  120    0  119
  120 G     119  121    0  120
  121 A     120  122    0  121
  ..........
```

Parameters:

	Input
Sequence	File with RNA sequence.
	Output
Result	Output file.
	Options
Minimal helix length	Minimal helix length. If specified, then given minimal helix length allowed. Minimal value is 2. Default value is 2.
Maximal distance	Maximal distance between begin and end of secondary structure. If specified, then given maximal secondary structure length allowed. Minimal value is 7, default value is 50.

BestPal-W

Program for searching best "linear" rna secondary structure for long sequences with a window moving along the sequence.

Method description.

A window with user-defined size moves along the sequence.

For each position of the window the best palindrome is calculated by dynamic programming method without "brahching" of structures.

Only the best variant goes to output file.

Output example:

```
FoldRNA Vienna format:
Length: 590 Energy: -70.1

        10        20        30        40        50        60
UAUUAUCGUGUGCAGUUAAAAUUGACUUUUUAAUGCGGCUCCAUUUUUGGGUCGGUGUUU
............................................................
        70        80        90       100       110       120
ACUAUUUGAUCAAGGGCUUAAAUAUUUUUGUCUUAAUACGAAAAAACGCACAGAUUUGGU
............................................................
       130       140       150       160       170       180
AAAGGCUUAACUUAAAAUUUCAGCGCCCAAUCACCCCCUUCAGAGUUGCCACACGUUGUU
............................................................
       190       200       210       220       230       240
ACACUAAGUUAUCGAAACGAACAGCUGAUUUUUGUUUUGUAAUAUUUGAGGUUGGUUUUU
............................................................
       250       260       270       280       290       300
GUUGGCUGAAAUAUUAUUACAUUAAAUUAGAUAUGGACCUUUUACUUCAAAGCGUUUGAC
............................................................
       310       320       330       340       350       360
AAGUUGAACAUCAAACGGAAAUCUAUUAUAGCCCCAAUUGGCGAGACCAUCAAAUAAUCA
..............................(((....((((.(((((((((......(((
       370       380       390       400       410       420
UUGGAAAACAACCUGAGAUGAGUUUUCCAGACAAGGCGGAGCGCAAAAAGUGCUGGAACA
(((.....)))..))))))).)))))))))....))).......................
       430       440       450       460       470       480
```

```
ACCGGGACGAGUAUUGGAAAUGUCUCGAGGAGCACGCCCCAAAGCACAGUUCUACCAGUG
............................................................
       490       500       510       520       530       540
GGGAAAAGGUACCAACCCCCUGCCAGAGUCUUCGCAAAUCAUUUGAGCAAUCCUGCCCUG
............................................................
       550       560       570       580       590
GUCAAUGGGUAAAGCACUUCGACCGCAAGCGUACUUAUGACCAGUUUAAG
..................................................

FoldRNA GCG format:
Length: 590 Energy: -70.1

    1 U       0    2    0    1
    2 A       1    3    0    2
    3 U       2    4    0    3
    4 U       3    5    0    4
    5 A       4    6    0    5
    6 U       5    7    0    6
    7 C       6    8    0    7
    8 G       7    9    0    8
    9 U       8   10    0    9
   10 G       9   11    0   10
   11 U      10   12    0   11
   12 G      11   13    0   12
   13 C      12   14    0   13
   14 A      13   15    0   14
   15 G      14   16    0   15
   16 U      15   17    0   16
   17 U      16   18    0   17
   18 A      17   19    0   18
   19 A      18   20    0   19
   20 A      19   21    0   20
   21 A      20   22    0   21
   22 U      21   23    0   22
   23 U      22   24    0   23
   24 G      23   25    0   24
   25 A      24   26    0   25
   26 C      25   27    0   26
   27 U      26   28    0   27
   28 U      27   29    0   28
   29 U      28   30    0   29
   30 U      29   31    0   30
   31 U      30   32    0   31
   32 A      31   33    0   32
   33 A      32   34    0   33
   34 U      33   35    0   34
   35 G      34   36    0   35
   36 C      35   37    0   36
   37 G      36   38    0   37
   38 G      37   39    0   38
   39 C      38   40    0   39
   40 U      39   41    0   40
...
```

Parameters:

Input	
Sequence	File with RNA sequence.
Output	
Result	Output file.
Options	
Window length	User-defined window size moving along the sequence. Window length does not exceed the input sequence length. Default value is 100, minimal value is 20, maximal

	value is 3000.

Find-miRNA

It is believed that most miRNAs are scarce in the cell and therefore are not yet discovered. The program FindMiRNA searches for miRNA genes and miRNAs within them.

The search procedure

The search process is conducted by successive filtering the genomic sequence. The procedure is organized in four steps: 1) fast estimation of secondary structure potential by calculation nucleotide scores; 2) search for hairpins and calculation of their energies; 3) estimation of thermodynamic probability of the hairpin structure found; 4) search for miRNAs in the candidate hairpin. In more details these filters are described below.

At first the FindMiRNA scans the input sequence with the sliding window of 100nt. Within the window it calculates nucleotide content and estimates E-score (the sequence potential to form stable secondary structure). It filters out the subsequences can not form the stable stable structures, i.e. which nucleotide content and E-score don not fall in the range of found miRNA genes. For clever filtering it takes into account the interdependency of nucleotide scores and interdependency of overlapping sequence windows. The step is the fastest one with time complexity of O(N).

At the second step FindMiRNA calls for another Softberry program, BestPal, which calculates the optimal imperfect hairpin which can be formed within a sequence window. The BestPal algorithm is based on the idea of dynamic programming realized in the wide-spread mfold algorithm for RNA secondary structure prediction. BestPal uses the energy parameters of Turner's energy rules. The hairpin energy is calculated summing over the energies of helixes and loops:

$$E_i = \sum_h e_h + \sum_l e_l$$

where e_h is helix energy and e_l is loop energy.

Searching for hairpins, BestPal omits secondary structure junctions and therefore works faster than Zuker's mfold program. Its time complexity is $O(N^{2.88})$ comparing with $O(N^{3.5})$ of mfold. When BestPal work is completed, the FindMiRNA saves the subsequences with stable hairpins only (free energy less than -17 kcal/mole by default). Though it takes most time, currently this step is the most effective in reducing the pre-miRNA candidate number.

At the third step FindMiRNA calls for RNAfold_bpp program. This filter takes the remaining sequences and calculates their matrices of base-pairing probabilities. The algorithm is based on McCaskill algorithm and dynamically calculates the partition function of RNA. Using partition function, our program calculates base-pairing probabilities of the ensemble of RNA structures. Using the optimal hairpin structure calculated at step 2, it estimates the hairpin probability and filters out the sequences with stable alternative structures. This step has the slowest time complexity of $O(N^{3.5})$, however, the initial sequence is already reduced by several orders at the steps 1 and 2.

At the final step FindMiRNA searches for miRNAs within the sequences remained. It calculates the weight matrix of any 21-mer oligonucleotide within a putative pre-miRNA and takes into account base-pairing characteristics of a candidate miRNA.

Currently the program is specially trained for three organisms (hsa, mmu and ath), although it can be used for others. We plan to extend the number of organisms analyzed and to automatically detect which of the analyzed genomes an input sequence belongs to.

Input and output

The program input is a genomic sequence and three-letter organism ID. The program outputs the putative pre-miRNAs and miRNAs in the following order:

- chain direction (+\-)
- the beginning and the end of a predicted pre-miRNA
- the beginning and the end of a predicted miRNA
- pre-miRNA sequence
- miRNA sequence

Parameters:

Input file - Input file

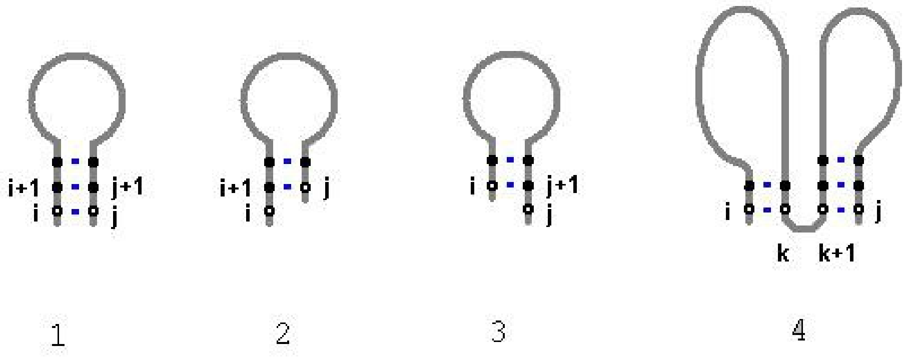
Output file - Output file

Window size - Scanning window size. Default value is 20, minimal value is 20, maximal value is 200.

Organism type - Organism type:
Homo Sapiens
Mus Musculus
Arabidopsis Thaliana

FoldRNA

Program for RNA secondary structure prediction based on dynamic programming (Nussinov and Jackonson, 1978, Zuker, 2005). For energy calculation nearest neighbor energy rules are used.
FoldRNA uses energy parameters similar to mfold.
FoldRNA uses energy parameters mainly from:
Turner D.H. and Sugimoto N. (1988) RNA structure prediction Ann.Rev.Biophys.Biophys.Chem. 17, pp. 167-92; Table 1
METHOD DESCRIPTION:
FoldRNA predicts optimal and suboptimal secondary structures of RNA using
dynamic algorithm for energy minimization.
Solution of a long sequence is decomposed into solutions of smaller problems:
Let's define E(i,j) = minimum energy for subchain starting at i and ending at j, and a(i,j) = energy of pair i,j.
If values E(i,j) are calculated for line which is maximally close to main diagonal of matrix LxL, where L = sequence length. (min. hairpin loop should have size not less than 3 nt), then we can find step by step this values for lines next after this, using the following recursion scheme (4 possible cases):



```
1) i,j is paired,      E(i,j)  = E(i+1,j-1) + a(i,j)
2) i is unpaired,      E(i,j)  = E(i+1,j  )
3) j is are unpaired,  E(i,j)  = E(i  ,j-1)
4) bifurcation         E(i,j)  = E(i,k) + E(k+1,j)
```

```
Recursion (iteration over length):

E(i,j)=min{
```

```
            E(i+1,j),

            E(i,j-1),

            E(i+1,j-1)+a(i,j),

            min  ( E(i,k) + E(k+1,j) )
           i<k<j

          }
```

When all matrix is filled, the programs searches for lowest value of E(i,j), and then restores by the matrix corresponding secondary structure and sends it to output.
Program is provided with viewer.

Output example:

```
Program RNAfold (Softberry Inc.) version 3.0
Sequence_name: "At-MIR156a_Stem" Length: 183

::: structure # 1 :::
Energy: -82.9 kkal/mol 75% in helices

        10        20        30        40        50        60
gugaaugaaagaguugggacaagagaaacgcaaagaaacugacagaagagagugagcaca
((((..((.((((((((((((((((((.....((((..(((((((((((((((((((((.
        70        80        90       100       110       120
caaaggcaauuugcauaucauugcacuugcuucucuugcgugcucacugcucuuucuguc
(((.(((((..((((......)))).)))))....))).))))))))).))).)))))))
       130       140       150       160       170       180
agauuccggugcugaucucuuuggccugucuucguucucuaugucucaaucucucucuau
))....(((.(((((.....))))))))))))..)))))).)))))))).)))).))..)
       190
cac
)))

GCG format:
    1 g       0    2  183    1
    2 u       1    3  182    2
    3 g       2    4  181    3
    4 a       3    5  180    4
    5 a       4    6    0    5
    6 u       5    7    0    6
    7 g       6    8  177    7
    8 a       7    9  176    8
    9 a       8   10    0    9
   10 a       9   11  174   10
   11 g      10   12  173   11
   12 a      11   13  172   12
   13 g      12   14  171   13
   14 u      13   15  169   14
   15 u      14   16  168   15
   16 g      15   17  167   16
   17 g      16   18  166   17
```

....

Parameters:

Input	
Sequence	File with RNA sequence.
Output	

Result	Output file.
	Options
Window size	Scanning window size. Default value is 20, minimal value is 20, maximal value is 200.
Organism type	Organism type: **Homo Sapiens** **Mus Musculus** **Arabidopsis Thaliana**

Target-miRNA

The program Target-miRna is developed for search for microRNA (miRNA) sites in genomic sequences. miRNAs promote mRNA cleavage at almost perfect complementarity to its site. In case of less complementarity, miRNAs inhibit mRNA translation. Our program Target-miRna searches a given target sequence for microRNA sites, basing on calculation of the interaction energy between miRNA and its site. Therefore Target-miRna can be used for search of both site types.

Target-miRna scans a target sequence and calculates the energy of complementary interaction between miRNA and possible site i as follows:

$$E_i = \sum_h e_h + \sum_l e_l$$

where e_h is helix energy and e_l is loop energy if any.

The energy parameters of complementary interactions and loops are taken from Turner's table. To skip suboptimal miRNA-site pairing we minimize the interaction energy by a dynamic algorithm which is based on Nussinov and Jackobson and Zuker papers. The user sets an energy threshold, and Target-miRna outputs all the candidate sites, which energy of miRNA-site interaction is lower (i.e., more stable) than it.

Target-miRna supports two different search modes. In the first mode the user inputs a single miRNA sequence by himself. In the second mode the user specifies the organism and our program searches for the sites for all miRNAs known for this organism, using built-in miRNA library. Currently the library contains the miRNAs of the following organisms:

cel (Caenorhabditis elegans)
hsa (Homo sapiens)
dme (Drosophila melanogaster)
mmu (Mus musculus)
ath (Arabidopsis thaliana)
rno (Rattus norvegicus)
oza (Oryza sativa)
ebv (Epstein Barr)
gga (Gallus gallus)
dps (Drosophila pseudoobscura)
dre (Danio rerio)
xla (Xenopus laevis)
zma (Zea mays)
sbi (Sorghum bicolor)
ame (Apis mellifera)
aga (Anopheles gambiae)
cfa (Canis familiaris)

Parameters:

Input	
Sequence	Name of the file with RNA sequence in FASTA format or just a sequence without a header.
Output	
Result	Filename for output (Vienna format, then GCG format).
Options	
Sequence Database	Genomic database of specific organism.
Energy threshold value	Energy threshold (default value is -25.0).

Repeats

LCRep

Program for mapping low complexity regions in nucleotide sequences.

Search for the low complexity regions is performed with using Shannon's information measure. Shannon's information is defiened as follows:

$$H=-\sum_{i=1}^{k} P(a_i)\log_2 P(a_i)$$

where: $\{a_1, ..., a_k\}$ is the alphabet of the size k, and $P(a_i)$ is a fractional composition of a_i

The search is carried out as follows. For each position i of the sequence S calculation of the Shannon's information $H(i, l)$ is performed in the window of size l within the range $[l_{begin}, l_{end}]$. If $H(i, l)$ turns out below prespecified threshold $H_{thr}(l)$ then fragment $[i, i+l]$ is declared low complex. Intersection of all such fragments at the end of calculation gives a map of low complexity regions of the sequence S.

Parameters:

<table>
<tr><th colspan="2">Input</th></tr>
<tr><td><b>Sequences set</b></td><td>Source file with nucleotide sequences in <b>multiFASTA</b> format Maximum file size is 1 GB.</td></tr>
<tr><th colspan="2">Output</th></tr>
<tr><td><b>Result</b></td><td>Name of the output file</td></tr>
<tr><td><b>Format</b></td><td>Result presentation mode examples:
• <b>Output list of low compl. repeat regions</b>
• &gt;c20
• Masked regions:
• p1: 90   p2: 115   l: 26   chain(+) [Low Complexity Region]
• p1: 220   p2: 240   l: 23   chain(+) [Low Complexity Region]
p1: - left position of Low Complexity Region
p2: - right position of Low Complexity Region
l: - length of Low Complexity Region
chain(+) - chain direction
• <b>Output sequence, masked lett. replaced with N</b>
• &gt;c20
• GCCAAGAAGATATGTAGCATTAAGGTTTAGAATACAGGCTTTGAAGTCAAACAGACCAGAGTTAACAACCTCATTTTGTT
• TTTATTTTCNNNNNNNNNNNNNNNNNNNNNNNNNNCTTTAAGTTCTAGGGTACATGTGCACAACGTGCAGGTTTGTTACA
• TATGTATACATGTGCCATGTTGGTGTGCTGCACCCATTAACTGGACATTTACATTAGGTNNNNNNNNNNNNNNNNNNNNN
• CCCTCCTCCCCTTACCCCACAACAGGCCCCGGTGTGTGATGTTCCCCTTCCTGTGTCCAAGTGTTCTCATTGTTCAGTTC
• <b>Output sequence, masked lett. are in upper case</b>
• &gt;c20
• gccaagaagatatgtagcattaaggtttagaatacaggctttgaagtcaaacagaccagagttaacaacctcattttgtt
• tttattttcTTTTTTAAAATTTTTTTAAAATTATActttaagttctagggtacatgtgcacaacgtgcaggtttgttaca
• tatgtatacatgtgccatgttggtgtgctgcacccattaactggacatttacattaggtAAAAAAAAAAAAAAAAAAAAA
• ccctcctccccttaccccacaacaggccccggtgtgtgatgttccccttcctgtgtccaagtgttctcattgttcagttc
•</td></tr>
</table>

Options	
Accurancy	Select one of the configuration files: **Normal** - default configuration **Sensitive** - more sensitive configuration resulting in higher masking percent **Rough** - more roung configuration resulting in lower masking percent

LCRrep-P

Program for mapping low complexity regions in protein sequences. Search for the low complexity regions is performed with using Shannon's information measure.

Search for the low complexity regions is performed with using Shannon's information measure. Shannon's information is defiened as follows:

$$H = -\sum_{i=1}^{k} P(a_i) \log_2 P(a_i)$$

where: $\{a_1, ..., a_k\}$ is the alphabet of the size k, and $P(a_i)$ is a fractional composition of a_i

The search is carried out as follows. For each position i of the sequence S calculation of the Shannon's information $H(i, l)$ is performed in the window of size l within the range $[l_{begin}, l_{end}]$. If $H(i, l)$ turns out below prespecified threshold $H_{thr}(l)$ then fragment $[i, i+l]$ is declared low complex. Intersection of all such fragments at the end of calculation gives a map of low complexity regions of the sequence S.

Parameters:

Input	
Sequences set	Source file with protein sequences in **multiFASTA** format Maximum file size is 1 GB.
Output	
Result	Name of the output file
Format	Result presentation mode examples: • **Output list of low compl. repeat regions** • `>EXAMPLE SEQ` • `Masked regions:` • `p1: 81        p2: 120       l: 40        chain(+)  [Low Complexity Region]` • `p1: 191       p2: 208       l: 18        chain(+)  [Low Complexity Region]` p1: - left position of Low Complexity Region p2: - right position of Low Complexity Region l: - length of Low Complexity Region chain(+) - chain direction • **Output sequence, masked lett. replaced with X** • `>EXAMPLE SEQ` • `ASFDPHEKQLIGDLWHKVDVAHCGGEALSRMLIVYPWKRRYFENFGDISNAQAIMHNEKVQAHGKKVLASFGEAVCHLDG` • `XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXIRAHFANLSKLHCEKLHVDPENFKLLGDIIIIVLAAHYPK` • `DFGLECHAAYQKLVRQVAAALAAEYHIGDLXXXXXXXXXXXXXXXXXX` • **Output sequence, masked lett. are in upper case**

	• >EXAMPLE SEQ • asfdphekqligdlwhkvdvahcggealsrmlivypwkrryfenfgdisnaqaimhnekvqahgkkvlasfgeavchldg • EEEEKKKKKEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEirahfanlsklhceklhvdpenfkllgdiiiivlaahypk • dfglechaayqklvrqvaaalaaeyhigdlEEEEEEEEEEEEEEEEEE
Options	
Accurancy	Select one of the configuration files: **Normal** - default configuration **Sensistive** - more sensitive configuration resulting in higher masking percent **Rough** - more roung configuration resulting in lower masking percent

MapRep

Finding and Mapping repeats from a given repeat database. Maps repeats on small genomes.

Parameters:

Input	
Genome	Name of input genome file
Repeat base	Name of input repeat base file (Multifasta in 4-letter alphabet)
Base	Select one of the configuration files: **Normal (slow)** **Rough (fast)**
Output	
Result	Name of output file
Format:	Output mode: **Repeat positions** **Mask repeats by symbol "N"** **Mask sequence. Sequence - lower case, Repeats - upper case** **Mask sequence. Sequence - upper case, Repeats - lower case**
Output string length	Output sequence string length.
Options	
Minimum repeat length	Minimum repeat length
Minimum repeat homology	Minimum repeat homology.
Minimum sum block	Minimum sum block repeat length in alignment
Minimum repeat number	Minimum repeat number for base entry.

TandemRep

Program for mapping the Tandem Repeats Regions in nucleotide sequences.

TandemRep mapping is performed by searching regions with uniform dinucleotide composition. The searching is initiated for the regions flanked by short ideal repeated elements.

Tandem searching algorithm consists of the following stages:

1) Find a pair of l-plets C_1 and C_2 with a distance between C_1 and C_2 not exceeding predefined N. The region between and including C_1 и C_2 will be denoted as R_1 with the length L_1. If C_1 and C_2 overlap then tandem unit size can be found trivially, jump to p.5.

2) Implying that C_1 and C_2 flanks do not contain insertions/deletions, extend synchronously C_1 and C_2 allowing 1 mismatch per several matches. Extended C_1 and C_2 we will denote as C_3 and

C_4. After this operation the region will be denoted as R_2 with the length L_2 (>= L_1). If extension performed without mismatches and C_3 and C_4 overlap then we have ideal tandem which unit size again can be found trivially, followed by jump to p.5. If extension performed with mismatches and C_3 and C_4 overlap then we have almost ideal tandem which unit size can be found according p.4. Proceed if C_3 and C_4 do not overlap.

3) Now region R_2 looks as follows

```
   C3                                   C4
########-----------------------------########
|  W1  |  W2  |  W3  |  W4  |  W...| Wn-1| Wn  |
```

For the region R_2 perform the following test. Divide region into set of windows W_1, …, W_n, each of size U. Consequently compare mono- (or di-) plet composition of the windows W_1 and W_i. If the difference in such composition between W_1 and some window W_i exceeds predefined threshold then stop. Test is not passed, jump to the p.1 to consider the next pair of l-plets. If the difference is low for all windows W_2, …, W_n then the test is passed and at least fragment R_2 could be declared tandem region.

Since we don't know the size of the window at which test described above could be passed, the test is performed for the window sizes U = 2, …, L_2/2.

Remember the lowest U at which the test is passed. Denote it U_1.

3a) Since uniform mono- (or di-) plet composition does not guarantee homology in windows W_1 and W_i, at this step the identity calculated by cycled Smith-Waterman algorithm is used for the additional filtering. If such an identity does not exceed predefined threshold then calculation is stopped for the C_1 and C_2 pair.

4) Calculate more precisely unit size U_{opt} of the tandem using two small windows synchronously sliding at the distance U one from another, U changes from U_1 to L_2/2.

5) Using U_{opt} calculated at the previous step find precise margins of the tandem using again two small synchronously sliding windows.

Such a procedure is carried out for all pairs C_1 and C_2 possible in the sequence. The final map of the tandems is an interception of tandems found for all l-plet pairs.

Parameters:

	Input
Sequences set	Source file with nucleotide sequences in **multiFASTA** format Maximum file size is 1 GB
Base	Select one of the configuration files: **Normal** - default configuration **Sensistive** - more sensitive configuration resulting in higher masking percent **Rough** - more roung configuration resulting in lower masking percent
	Output
Result	Name of the output file
Format	Result presentation mode examples: - **Output list of tandem repeat regions**

- `>c20`
- `Masked regions:`
- `p1:  96       p2: 127      l: 31         chain(+) [Tandem Repeat]`
- `p1: 240       p2: 262      l: 22         chain(+) [Tandem Repeat]`
- `p1: 277       p2: 322      l: 45         chain(+) [Tandem Repeat]`

  p1: - start position of the tandem region
  p2: - end position of the tandem region
  l: - length of the tandem region
  chain(+) - chain direction

- **Output sequence, masked lett. replaced with N**
- `>c20`
- `CGGTGGCGGCAGCCGGCTCAAGCCCGGGCCGCAGCTGCCTGGCCGCGGGGGCCGCCGAGCAGCGGGAGGGCCTTTGGGGG`
- `GCGGGGCGGCGGCGCCNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNCTGCTGGAAGTGCGCCTGGTCGAGACCCCGGGG`
- `CGGGAGCTGTGGAGGATGGTCCCGGCGGGACGGGCCGCTCGGGGACAAGCGGAGCGCGCCCAAGGGCCGTCGGGCGAGGG`
- `NNNNNNNNNNNNNNNNNNNNNNTCCCCGACACCGTCNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN`
- `NTGAGGAAGGTGAAGAGGAGACGGTGGCGTCGGGGGAGGAGTCGCTGGGCTTTCTGTCCAGGCTGCCCCCTGGCCCGGCC`
- **Output sequence, masked lett. are in upper case**
- `>c20`
- `cggtggcggcagccggctcaagcccgggccgcagctgcctggccgcgggggccgccgagcagcgggagggcctttggggg`
- `gcggggcggcggcgccCGAGGACGACGATGAAGACGACGACGAGGAGctgctggaagtgcgcctggtcgagaccccgggg`
- `cgggagctgtggaggatggtcccggcgggacgggccgctcggggacaagcggagcgcgcccaagggccgtcgggcgaggg`
- `GGCGGCCGCCGCCGCCGCCGCCtccccgacaccgtcGGAGGACGAGGAGCCGGAGGAAGAGGAGGAGGAGGCGGCAGCGG`
- `Ctgaggaaggtgaagaggagacggtggcgtcgggggaggagtcgctgggctttctgtccaggctgccccctggcccggcc`
- **Output repeats during calculation (regions may overlap)**
- `>seq:1  beg:96  len:31`
- `CGAGGACGACGATGAAGACGACGACGAGGAG`
- `>seq:1  beg:240  len:22`
- `GGCGGCCGCCGCCGCCGCCGCC`
- `>seq:1  beg:277  len:45`
- `GGAGGACGAGGAGCCGGAGGAAGAGGAGGAGGAGGCGGCAGCGGC`

  seq:1 - sequence number in input file
  beg: - start position of the Tandem Repeat
  len: - length of the Tandem Repeat

Options	
Minimal length	Lowest acceptable tandem region length
Maximum diplet distance	Maximum acceptable difference in diplet composition between two windows in the tested region (from 0 to 200)
Maximum unit size	Maximum acceptable tandem unit size
Smith-Waterman identity	Minimum allowed identity in Smith-Waterman algorithm for repeated units
Strict extending	Extend tandem with more strict conditions for shorter units and low monoplet complexity regions.

TandemRep-P

Program for mapping the Tandem Repeats Regions in protein sequences.

TandemRep mapping is performed by searching regions with uniform dinucleotide composition. The searching is initiated for the regions flanked by short ideal repeated elements.

Tandem searching algorithm consists of the following stages:

1) Find a pair of l-plets C_1 and C_2 with a distance between C_1 and C_2 not exceeding predefined N. The region between and including C_1 и C_2 will be denoted as R_1 with the length L_1. If C_1 and C_2 overlap then tandem unit size can be found trivially, jump to p.5.

2) Implying that C_1 and C_2 flanks do not contain insertions/deletions, extend synchronously C_1 and C_2 allowing 1 mismatch per several matches. Extended C_1 and C_2 we will denote as C_3 and C_4. After this operation the region will be denoted as R_2 with the length L_2 (>= L_1). If extension performed without mismatches and C_3 and C_4 overlap then we have ideal tandem which unit size again can be found trivially, followed by jump to p.5. If extension performed with mismatches and C_3 and C_4 overlap then we have almost ideal tandem which unit size can be found according p.4 (jump to p.4). Proceed if C_3 and C_4 do not overlap.

3) Now region R_2 looks as follows

```
   C3                                   C4
########-----------------------------########
| W1  | W2  | W3  | W4  | W...| Wn-1| Wn  |
```

For the region R_2 perform the following test. Divide region into set of windows W_1, …, W_n, each of size U. Consequently compare mono- (or di-) plet composition of the windows W_1 and W_i. If the difference in such composition between W_1 and some window W_i exceeds predefined threshold then stop. Test is not passed, jump to the p.1 to consider the next pair of l-plets. If the difference is low for all windows W_2, …, W_n then the test is passed and at least fragment R_2 could be declared tandem region.

Since we don't know the size of the window at which test described above could be passed, the test is performed for the window sizes U = 2, …, $L_2/2$.

Remember the lowest U at which the test is passed. Denote it U_1.

3a) Since uniform mono- (or di-) plet composition does not guarantee homology in windows W_1 and W_i, at this step the identity calculated by cycled Smith-Waterman algorithm is used for the additional filtering. If such an identity does not exceed predefined threshold then calculation is stopped for the C_1 and C_2 pair.

4) Calculate more precisely unit size U_{opt} of the tandem using two small windows synchronously sliding at the distance U one from another, U changes from U_1 to $L_2/2$.

5) Using U_{opt} calculated at the previous step find precise margins of the tandem using again two small synchronously sliding windows.

Such a procedure is carried out for all pairs C_1 and C_2 possible in the sequence. The final map of the tandems is an interception of tandems found for all l-plet pairs.

Parameters:

Input	
Sequences set	Source file with nucleotide sequences in **multiFASTA** format Maximum file size is 1 GB
Base	Select one of the configuration files: **Normal** - default configuration **Sensistive** - more sensitive configuration resulting in higher masking percent **Rough** - more roung configuration resulting in lower masking percent
Output	
Result	Name of the output file
Format	Result presentation mode examples: • **Output list of tandem repeat regions** • `>EXAMPLE SEQ` • `Masked regions:` • `p1:  81       p2: 120       l: 40         chain(+)  [Tandem Repeat]` • `p1: 191       p2: 208       l: 18         chain(+)  [Tandem Repeat]` p1: - left position of the Tandem Repeat p2: - right position of the Tandem Repeat l: - length of the Tandem Repeat chain(+) - chain direction • **Output sequence, masked lett. replaced with X** • `>EXAMPLE SEQ` • `ASFDPHEKQLIGDLWHKVDVAHCGGEALSRMLIVYPWKRRYFENFGDISNAQAIMHNEKVQAHGKKVLASFGEAVCHLDG` • `XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXIRAHFANLSKLHCEKLHVDPENFKLLGDIIIIVLAAHYPK` • `DFGLECHAAYQKLVRQVAAALAAEYHIGDLXXXXXXXXXXXXXXXXXX` • **Output sequence, masked lett. are in upper case** • `>EXAMPLE SEQ` • `asfdphekqligdlwhkvdvahcggealsrmlivypwkrryfenfgdisnaqaimhnekvqahgkkvlasfgeavchldg` • `EKEKEKEKEKEKEKEKEKEKEEKEKEKEKEKEKEKEKEKEirahfanlsklhceklhvdpenfkllgdiiiivlaahypk` • `dfglechaayqklvrqvaaalaaeyhigdlEKPEKPEKPEKPEKPEKP` • **Output repeats during calculation (regions may overlap)** • `>seq:1  beg:81  len:40` • `EKEKEKEKEKEKEKEKEKEKEEKEKEKEKEKEKEKEKEKE` • `>seq:1  beg:191  len:18` • `EKPEKPEKPEKPEKPEKP` seq:1 - sequence number in input file beg: - start position of the Tandem Repeat len: - length of the Tandem Repeat
Options	
Minimal length	Lowest acceptable tandem region length
Maximum diplet distance	Maximum acceptable difference in diplet composition between two windows in the tested region (from 0 to 200)
Maximum	Maximum acceptable tandem unit size

unit size	
Smith-Waterman identity	Minimum allowed identity in Smith-Waterman algorithm for repeated units
Strict extending	Extend tandem with more strict conditions for shorter units and low monoplet complexity regions.

FindRep

Find repeats and create prior repeats base.

SelTag

Data specification

The expression data for the set of genes is represented as a table, consisting of rows (usually corresponding to genes) and columns (or fields, usually corresponding to samples/tissues/experiments). Each row corresponds to expression measurements for the gene. Columns correspond to experiments/samples/tissues. However, this table may include not only expression data, but also other information related to genes, for example gene names, classifiers, etc. Therefore we will call the table columns as 'fields' in general case. In general, columns of the table could be of four basic types:

IVALUE	signed integer value;
FVALUE	floating point value;
WORD	text without spaces inside (single word);
STRING	text with spaces inside allowed.

Fields are completely defined by their basic types and names.

SelTag Input file basic format

Basic input file format should be as follows:

```
; May contain comment starting from the semicolon in any line of the file
NAME<tab>WORD
GENEID<tab>IVALUE
TISSUECANCER0<tab>FVALUE
TISSUECANCER1<tab>FVALUE
TISSUENORMAL0<tab>FVALUE
TISSUENORMAL1<tab>FVALUE
TISSUENORMAL2<tab>FVALUE
#GROUP<tab>Cancer tissues
TISSUECANCER0
TISSUECANCER1
#ENDGROUP
#GROUP<tab>Arbitrary group
TISSUECANCER1
TISSUECANCER2
TISSUENORMAL0
TISSUENORMAL1
#ENDGROUP
END
DATA
GENE04675<tab>402<tab>6.00<tab>5.60<tab>5.97<tab>6.00<tab>6.00
GENE46890<tab>794<tab>2.77<tab>3.22<tab>5.65<tab>5.68<tab>5.68
GENE23794<tab>404<tab>5.97<tab>5.97<tab>6.00<tab>5.60<tab>5.97
```

In this example <tab> implies 'Tab' character symbol.

First lines (up to the "DATA" line) contain data format description. In this part of the file each line describes field description: field name and field basic type.

After the "DATA" line - data on each gene are represented. Each line correspond single cards. Field data are separated by 'tab' symbol. Double 'tab' is interpreted as missed data.

It is assumed in SetTag program that the expression data in the file are normalized and the expression levels of genes in experiments are comparable.

Selection files.

MolQuest version of the SelTag program can also operates with other types of files, namely, selection files. These files contain information about some selected genes or samples from the large data file in SelTag format. The selection file contain: the data file name from which selection was obtained; type of selection data (genes of samples), list of selected objects (their indices in the large data file). The selection files are in the XML format. Two examples are below.

Selection for some genes.

```
<?xml version="1.0" encoding="ISO-8859-5"?>
<SELECTION>
        <HEADER name="cc_Selection5">
                <DATA source="c:/data/cc.txt"/>
                <COMMENT><![CDATA["$F1 == "GEN14263" | $F12 >= 300"]]></COMMENT>
        </HEADER>
        <ELEMENTS type="GENES" count="9">
        <![CDATA[0;1;2;10;14;15;17;26;30]]>
        </ELEMENTS>
</SELECTION>
```

Selection for some fields (samples).

```
<?xml version="1.0" encoding="ISO-8859-5"?>
<SELECTION>
        <HEADER name="notterman2001_set1">
                <DATA source="c:/data/notterman2001_set1.txt"/>
                <COMMENT><![CDATA["From cc.txt data file."]]></COMMENT>
        </HEADER>
        <ELEMENTS type="FIELDS" count="10">
          <![CDATA[0;1;2;3;5;6;7;18;19;30]]>
        </ELEMENTS>
</SELECTION>
```

Selection files may be selected during the SelTag execution and also used by SelTag for calculation and/or visualization. Note, each selection file is linked to large data file by its name. Selection data cannot be applied to another data file.

BdClust

Clustering of gene expression profiles or samples by Ben-Dor algorithm.

Algorithm description

The program allows clustering genes by their expression profile similarity. The purpose of the analysis is to select groups of genes that have common patterns of expression in different experiments, e.g. high expression in cancer tissues and low expression in normal tissues. These patterns of co-expression are usually treated as co-regulation. The similarity of the expressions patterns may not be limited by simple rules and can be described by similarity (or distance) Measures. There are several measures of expression profile similarity between two genes:

(1) *Euclidean distance.* This is the geometric distance in the multidimensional space. It is computed as: $d_{ij} = [\Sigma_k (x_{ik} - x_{ik})^2]^S$, where x_i, x_j are two expression profiles for genes i,j , k is the index of experiment (field), x_{ik} is the expression value of gene i in the experiment k.

(2) *Squared Euclidean distance.* The squared Euclidean distance can be implemented in order to place progressively greater weight on objects that are further apart. The squared Euclidian distance is computed as: $d_{ij} = \Sigma_k (x_{ik} - x_{ik})^2$ (see explanation above). The Euclidian and squared Euclidian distances are computed from raw data (non-standardized), therefore they may be affected by differences in scale among the expression values in different experiments.

(3) *Manhattan distance.* This distance is the average absolute difference for the set of experiments calculated by the formula $d_{ij} = \Sigma_k |x_{ik} - x_{ik}|$. In most cases, this distance measure yields results similar to the simple Euclidean distance, for this measure, the effect of single large differences is dampened (since they are not squared).

(4) *Chebychev distance.* This distance is computed as $d_{ij} = \max_k | x_{ik} - x_{ik} |$. The measure is useful when one wants to define two objects as "different" if they are different on any one of the experiments.

In SelTag all distance measures (1-3) are normalized to the number of fields involved in calculation. This is useful when take into account expression data with missing values.

Other measures involve correlation coefficient r_{ij} between two expression profiles of genes *i* and *j*.

(5) $1-r_{ij}$; This measure keep close profiles with positive correlation coefficients and is useful when one wants to detect co-regulated genes.

(6) $1-|r_{ij}|$; This measure keep close profiles with higher absolute value of correlation coefficients.

(7) $1+|r_{ij}|$; This measure keep close profiles with negative value of correlation coefficients (anti-correlated).

Three types of correlation are possible for correlation distance option:
Pearson's *r* - Pearson's correlation coefficient. The Pearson product moment correlation coefficient between expression profiles *i* and *j* is calculated as follows:

$$r_{ij} = \frac{\sum_k (y_{ki} - \bar{y}_i)(y_{kj} - \bar{y}_j)}{\left(\sum_k (y_{ki} - \bar{y}_i)^2 \sum_k (y_{kj} - \bar{y}_j)^2\right)^{1/2}} ,$$

where y_{ki} is the expression level of gene *i* in the experiment *k*; $\bar{y}_i$ is the mean expression level of the gene *i*. Positive correlation implies that the expression levels of genes *i,j* are related positively, the higher expression of gene *i*, the higher expression of gene *j*. Negative correlation means that the expression levels of genes *i,j* are related negatively, the higher expression of gene *i*, the lower expression of gene *j*. If the r_{ij} is close to zero, two expression profiles are unrelated.
Spearman *r* - Spearman's correlation coefficient.
This correlation coefficient is computed for ranks. Let R_{ki} is the rank of the expression level in the experiment *k* of gene *i* (relatively to other experiments), R_{kj} is the rank of the expression level in the experiment *k* of gene *j*. Then Spearman's correlation coefficient is calculated by the formula

$$r_{ij} = \frac{\sum_k (R_{ki} - \bar{R}_i)(R_{kj} - \bar{R}_j)}{\left(\sum_k (R_{ki} - \bar{R}_i)^2 \sum_k (y_{kj} - \bar{R}_j)^2\right)^{1/2}}$$

Kendall's τ - Kendall's *tau* correlation coefficient.
To calculate Kendall's τ K for data points (y_{ki}; y_{kj}) 2K(K − 1) pairs considered (without self-pairing, the points in either order count as one pair). Pairs in which $y_{ki} > y_{mi}$ and $y_{kj} > y_{mj}$ or $y_{ki} < y_{mi}$ and $y_{kj} < y_{mj}$ are called concordant pairs (agreement between ranks), pairs with rank disagreement are called discordant pairs. In general, τ is calculated as

τ = ([number of concordant] – [number of discordant]) / total number of pairs

Clustering algorithm

The program implements Cluster Affinity Search Technique (CAST), proposed by Ben-Dor et al [Ben-Dor A., Shamir R., Yakhini Z. (1999) *J. Comput. Biol.* 6, 281–297].

A common shortcoming of hierarchical clustering techniques, such as single-linkage, complete-linkage, group-average, and centroid, is due to their "greedy" nature, once a decision to join two elements in one cluster is made, it cannot be undone. The CAST algorithm use the "affinity" values to perform "cleaning" step while making clusters by removing low-affinity elements of the cluster. The affinity in the CAST algorithm is the average similarity between gene expression profile and gene profiles already included to the cluster. The threshold for affinity is user-defined.

Example of output data

```
status=Correlation matrix calculation...
status=CAST clustering...
status=done [0.0 sec]
Number of gene clusters obtained 4.
Cluster Sizes and Scores:
Cluster 1      2      1.7469
Cluster 2      10     1.6321
Cluster 3      7      1.7248
Cluster 4      4      1.6679
List of selected genes, their cluster indices and scores :
No      DataIndex      Name    Cluster Score
1       1      GEN30482       2      1.6892
2       2      GEN03437       2      1.6962
3       3      GEN03687       2      1.6649
4       4      GEN24649       2      1.6463
```

Some lines starting from "`status=`" are just output the status of the calculation and can be ignored. Then the result cluster information is output: number of clusters, their list with cluster scores. Then list of selected genes with their cluster indices and scores is printed out.

Parameter description:

Input	
Expression data	Input file in seltag format

Fields select	**List of fields** - List of expression fields (tissues) used to calculate correlation between gene expression profiles, namely field indices in data format of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input:
	1;2;3-7;12;
	1-12;
	Selection data - Filename for fields selection in XML format. This is another way to set the list of fields.

Genes for select	**Genes for select** - List of genes to calculate correlation, namely gene indices in data set of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input:
	1;2;3-7;12;
	1-12;
	Gene list - Filename for genes selection in XML format for Gene List 1. This is another way to set the list of genes.

Output	
Result	Name of output file
Options	
Select clustering	Select clustering objects: genes or samples.

objects	
Type of distance	Type of distance between expression profiles. Several types of correlations are possible: 1-r_{ij}; 1-|r_{ij}|; 1+r_{ij}; Squared Euclidian distance; Euclidian distance; Manhattan distance; Chebyshev distance.
Type of correlation	Type of correlation coefficient. Three types of correlations are possible: Pearson's r, Spearman rank correlation and Kendall *tau* correlation.
Type of distance threshold	Type of distance threshold for clustering: **User-specified** **Average distance**
Threshold Value	The value of threshold, if user-specified type is set.
Clustering speed	This parameter set clustering speed: **Fast** mode stores distance matrix in memory (needs more memory for large data), **Slow** mode recalculates distance between gene pair (no memory limitations, appropriate for very large data).
Missing data treatment	Option to treat missing data. Several options are possible: **Substitute by means** (missing data are substituted by expression means in corresponding field); **Case-wise deletion** (correlations/distances are calculated by excluding cases that have missing data for any of the selected variables, all correlations are based on the same set of data); **Pair-wise deletion** (correlations/distances between each pair of profiles are calculated from all fields/samples having valid data for those two profiles).

CHPImport

Import expression data from the Affymetrix CHP format to SelTag data file.

Data specification

The input for **CHPImport** is the set of expression data in Affymetrix CHP data format, corresponding CDF file and file with list of CHP files to be processed and their short description (this file is provided by user). The CHP data already processed by statistical algorithm. The output is SelTag data file with gene expression data.

The program can read a set of CHP data files for the same chip. The output file is in **Seltag** format and reports the #HEADER section: Experiment filename; Algorithm name, DataHeader as reported in the CEL file, DataScalingFactor (*sf* value), DataNormalizationFactor (*nf* value), DataSignalTrimmedMean.

Example of experiment list file

```
GSM42890        DEHP_48hr_Veh1  DEHP 48hr Veh1
GSM42891        DEHP_48hr_Veh2  DEHP 48hr Veh2
GSM42892        DEHP_48hr_Veh3  DEHP 48hr Veh3
GSM42893        DEHP_48hr_Veh4  DEHP 48hr Veh4
GSM42894        DEHP_48hr_Veh5  DEHP 48hr Veh5
```

This file contains three columns separated by symbol. First column is the experiment data name (the corresponding CEL file should start from this name and have extension *.chp, for example GSM42890.chp). Second column is the name of the variable in the output SelTag file, corresponding to this experiment (see below example of SelTag output file). This column should not contain spaces. Third column is the extended description of the experiment that will appear at the SelTag file header section.

Example of output data

```
#HEADER
Import expression data from the set of CHP files.
  1 ExperimentDataFilename=GSM42883.cel
  1 DataHeader=Clof_168hr_t    Clof 168hr treated POOLED
  1 Algorithm name:ExpressionStat
  1 Algorithm parameters:BF= Alpha1=0.04 Alpha2=0.06 Tau=0.015 Gamma1H=0.0025
Gamma1L=0.0025 Gamma2H=0.003 Gamma2L=0.003 Perturbation=1.1 TGT=1500
NF=1.000000 SF=29.560343 SFGene=All
  1 Algorithm summary:Background=Avg:29.82,Stdev:1.12,Max:32.6,Min:27.2
Noise=Avg:1.02,Stdev:0.05,Max:1.2,Min:0.9 RawQ=0.98
  1 Algorithm ver:5.0
  1 Program:GeneChipAnalysis.GEBaseCall.1
  1 Probe array type:RG_U34A
#ENDHEADER
ProbesetName   STRING
Clof_168hr_t_Signal    FVALUE
Clof_168hr_t_Detection WORD
Clof_168hr_t_Detection_p       FVALUE
END
DATA
AFFX-MurIL2_at 37.5396 A       0.78955
AFFX-MurIL10_at51.8929 A       0.60308
AFFX-MurIL4_at 5.7568  A       0.97607
AFFX-MurFAS_at 32.2922 A       0.60308
AFFX-BioB-5_at 714.0201        A       0.08359
AFFX-BioB-M_at 1563.2017       P       0.00125
AFFX-BioB-3_at 800.5414        P       0.00359
AFFX-BioC-5_at 3686.6155       P       0.00017
AFFX-BioC-3_at 1989.3492       P       0.00006
AFFX-BioDn-5_at2807.6296       P       0.00066
AFFX-BioDn-3_at16410.8984      P       0.00020
AFFX-CreX-5_at 32975.3750      P       0.00004
```

Parameter description:

Input	
CDF file	The name of the CDF file for experiment set.
CHP directory	The name of the directory where all *.chp files can be found.
Experiment list file	File with experiment list and their description included into calculation.
Output	
Result	File with the resulting gene expression data in SelTag format.
Options	
Signal Only	If this flag set on, only signal values will be at the output. Otherwise, detection and detection p-values will be reported also.

FieldCorr

The program calculates correlation coefficients between the gene expression values in experiments (fields).

Program description

User should define two lists of fields; program will calculate correlation coefficients between gene expression values at the fields (samples) from different lists. User can also set the threshold for correlation value to select most correlated pairs of fields. The correlation coefficient is calculated for all genes available.

Three types of correlation are possible:

<u>Pearson's *r*</u> - Pearson's correlation coefficient. The Pearson product moment correlation coefficient between expression profiles *i* and *j* is calculated as follows:

$$r_{ij} = \frac{\sum_k (y_{ki} - \bar{y}_i)(y_{kj} - \bar{y}_j)}{(\sum_k (y_{ki} - \bar{y}_i)^2 \sum_k (y_{kj} - \bar{y}_j)^2)^{1/2}} \quad ,$$

where y_{ki} is the expression level of gene *i* in the experiment *k*; $\bar{y}_i$ is the mean expression level of the gene *i*. Positive correlation implies that the expression levels of genes *i,j* are related positively, the higher expression of gene *i*, the higher expression of gene *j*. Negative correlation means that the expression levels of genes *i,j* are related negatively, the higher expression of gene *i*, the lower expression of gene *j*. If the r_{ij} is close to zero, two expression profiles are unrelated.

<u>Spearman *r*</u> - Spearman's correlation coefficient.

This correlation coefficient is computed for ranks. Let R_{ki} is the rank of the expression level in the experiment *k* of gene *i* (relatively to other experiments), R_{kj} is the rank of the expression level in the experiment *k* of gene *j*. Then Spearman's correlation coefficient is calculated by the formula

$$r_{ij} = \frac{\sum_k (R_{ki} - \bar{R}_i)(R_{kj} - \bar{R}_j)}{(\sum_k (R_{ki} - \bar{R}_i)^2 \sum_k (y_{kj} - \bar{R}_j)^2)^{1/2}}$$

<u>Kendall's τ</u> - Kendall's *tau* correlation coefficient.

To calculate Kendall's τ K for data points (y_{ki}; y_{kj}) 2K(K − 1) pairs considered (without self-pairing, the points in either order count as one pair). Pairs in which $y_{ki} > y_{mi}$ and $y_{kj} > y_{mj}$ or $y_{ki} < y_{mi}$ and $y_{kj} < y_{mj}$ are called concordant pairs (agreement between ranks), pairs with rank disagreement are called discordant pairs. In general, τ is calculated as

τ = ([number of concordant] – [number of discordant]) / total number of pairs

Example of the output data

```
Correlation coefficients (Spearman rank correlation) between field expression data:
FieldList1\FieldList2 BC_1_tum       BC_1_tum0      BC_3_tum       BC_4_met
BC_1_tum0       0.4507 1.0000 0.5710 0.7502
BC_5_met        0.7135 0.7354 0.4533 0.8437
BC_6_tum        0.6044 0.7008 0.4573 0.8303
BC_7_tum        0.5856 0.3001 0.5085 0.3592
BC_8_met        1.0000 0.4507 0.2643 0.5407
BC_9_tum        0.8076 0.4445 0.4591 0.3603
List  of  gene  pairs  with  the  absolute  value  of  the  correlation  coefficients  above  threshold
(0.8076)
BC_5_met        BC_4_met        :       0.8437
BC_6_tum        BC_4_met        :       0.8303
BC_8_met        BC_1_tum        :       1.0000
BC_9_tum        BC_1_tum        :       0.8076
```

First line is the header. It contains the type of the calculated correlation in parentheses. Second line is the list if field names from the List1, separated by tabulation. Next lines list data for fields for List2 separated by tabulation.

Parameter description:

Input	
SelTag data	Input file in seltag format
Fields select	**List of fields** - List of fields to calculate correlation, namely field indices in data format of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input: 1;2;3-7;12;

	1-12;
	ALL;
	Fields list - Filename for fields selection 1 in XML format. This is another way to set the list of fields.

Fields select	**List of fields** - List of expression fields (tissues) used to calculate correlation between gene expression profiles, namely field indices in data format of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input: 1;2;3-7;12; 1-12; ALL **Fields list** - Filename for fields selection 2 in XML format. This is another way to set the list of fields.
Output	
Result	Name of output file
XML data	Name of the file for graphical output of correlation coefficient value profiles. If not specified then no graph output assumed.
Title	User-specified title of the graph plot.
Author	User-specified name of the graph author.
Comment	User-specified graph additional commentary line.
X axis name	User-specified graph X axis name.
Y axis name	User-specified graph Y axis name.
Options	
Type of correlation	Type of correlation coefficient. Three types of correlations are possible: Pearson's r, Spearman rank correlation and Kendall *tau* correlation.
Correlation threshold type	Type of threshold to select best correlating gene pairs. Several options are possible: Best N correlations ; Best % correlations; Correlation coefficient value; Select all pairs.
Correlation threshold value	Threshold to select genes from List 1 on the basis of the their correlation coefficient value to genes from List 2.
Missing data treatment	Option to treat missing data. Several options are possible : Substitute by means (missing data are substituted by expression means in corresponding field); Case-wise deletion (correlations/distances are calculated by excluding cases that have missing data for any of the selected variables, all correlations are based on the same set of data); Pair-wise deletion (correlations/distances between each pair of profiles are calculated from all fields/samples having valid data for those two profiles).

GeneCorr

The program calculates correlation coefficients between the gene expression profiles.
Program description

User should define two lists of genes, program will calculate correlation coefficients between gene expression profiles from different lists. User can also set the threshold for correlation value to select most correlated pairs.

User should provide list of fields to calculate correlation.
Three types of correlation are possible:

Pearson’s *r* - Pearson's correlation coefficient. The Pearson product moment correlation coefficient between expression profiles *i* and *j* is calculated as follows:

$$r_{ij}=\frac{\sum_{k}(y_{ki}-\bar{y}_i)(y_{kj}-\bar{y}_j)}{(\sum_{k}(y_{ki}-\bar{y}_i)^2\sum_{k}(y_{kj}-\bar{y}_j)^2)^{1/2}} ,$$

where y_{ki} is the expression level of gene *i* in the experiment *k*; $\bar{y}_i$ is the mean expression level of the gene *i*. Positive correlation implies that the expression levels of genes *i,j* are related positively, the higher expression of gene *i*, the higher expression of gene *j*. Negative correlation means that the expression levels of genes *i,j* are related negatively, the higher expression of gene *i*, the lower expression of gene *j*. If the r_{ij} is close to zero, two expression profiles are unrelated.

Spearman *r* - Spearman's correlation coefficient.

This correlation coefficient is computed for ranks. Let R_{ki} is the rank of the expression level in the experiment *k* of gene *i* (relatively to other experiments), R_{kj} is the rank of the expression level in the experiment *k* of gene *j*. Then Spearman’s correlation coefficient is calculated by the formula

$$r_{ij}=\frac{\sum_{k}(R_{ki}-\bar{R}_i)(R_{kj}-\bar{R}_j)}{(\sum_{k}(R_{ki}-\bar{R}_i)^2\sum_{k}(y_{kj}-\bar{R}_j)^2)^{1/2}}$$

Kendall’s τ - Kendall's *tau* correlation coefficient.

To calculate Kendall’s τ K for data points (y_{ki}; y_{kj}) 2K(K − 1) pairs considered (without self-pairing, the points in either order count as one pair). Pairs in which $y_{ki} > y_{mi}$ and $y_{kj} > y_{mj}$ or $y_{ki} < y_{mi}$ and $y_{kj} < y_{mj}$ are called concordant pairs (agreement between ranks), pairs with rank disagreement are called discordant pairs. In general, τ is calculated as

τ = ([number of concordant] – [number of discordant]) / total number of pairs

Example of the output data

```
Correlation coefficients (Spearman rank correlation) between gene expression profiles:
List1\List2     GEN30482        GEN03437        GEN30823
GEN01998        0.5657  0.4885  0.4939
GEN03687        0.7642  0.7814  0.7617
GEN24649        0.5858  0.5624  0.6399
GEN09108        0.1657  0.0949  -0.1042
GEN09514        0.4313  0.3925  0.2861
GEN02303        0.5876  0.5993  0.4568
List of gene pairs with the absolute value of the correlation coefficients above threshold
(0.7722)
GEN03687        GEN03437        :       0.7814
GEN02374        GEN03437        :       0.7941
GEN02374        GEN30823        :       0.8520
```

First line is the header. It contains the type of the calculated correlation in parentheses. Second line is the list if gene identifiers from the List1, separated by tabulation. Next lines list data for genes for List2 separated by tabulation.

Parameter description:

Input	
Expression data	Input file in seltag format
Fields select	**List of fields** - List of expression fields (tissues) used to calculate correlation between gene expression profiles, namely field indices in data format of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input: 1;2;3-7;12;

	1-12; **Selection data** - Filename for fields selection in XML format. This is another way to set the list of fields.

Genes for select	**List 1 of genes** - List of genes to calculate correlation, namely gene indices in data set of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input:
	1;2;3-7;12;
	1-12;
	Gene list 1 - Filename for genes selection in XML format for Gene List 1. This is another way to set the list of genes.

Genes for comparison	**List 2 of genes** - List of genes to calculate correlation, namely gene indices in data set of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input:
	1;2;3-7;12;
	1-12;
	Gene list 2 - Filename for genes selection in XML format for Gene List 2. This is another way to set the list of genes.

Output	
Result	Name of output file
XML data	Name of the file for graphical output of correlation coefficient value profiles. If not specified then no graph output assumed.
Title	User-specified title of the graph plot.
Author	User-specified name of the graph author.
Comment	User-specified graph additional commentary line.
X axis name	User-specified graph X axis name.
Y axis name	User-specified graph Y axis name.
Options	
Type of correlation	Type of correlation coefficient. Three types of correlations are possible: Pearson's r, Spearman rank correlation and Kendall *tau* correlation.
Correlation threshold type	Type of threshold to select best correlating gene pairs. Several options are possible: Best N correlations ; Best % correlations; Correlation coefficient value; Select all pairs.
Correlation threshold value	Threshold to select genes from List 1 on the basis of the their correlation coefficient value to genes from List 2.
Missing data treatment	Option to treat missing data. Several options are possible : Substitute by means (missing data are substituted by expression means in corresponding field); Case-wise deletion (correlations/distances are calculated by excluding cases that have missing data for any of the selected variables, all correlations are based on the same set of data); Pair-wise deletion (correlations/distances between each pair of profiles are calculated from all fields/samples having valid data for those two profiles).

HClust

The program allows clustering genes by their expression profile similarity. The purpose of the analysis is to select groups of genes that have common patterns of expression in different experiments, e.g. high expression in cancer tissues and low expression in normal tissues. These patterns of co-expression are usually treated as co-regulation. The similarity of the expressions patterns may not be limited by simple rules and can be described by similarity (or distance) Measures. There are several measures of expression profile similarity between two genes:

(1) *Euclidean distance.* This is the geometric distance in the multidimensional space. It is computed as: $d_{ij} = [\Sigma_k (x_{ik} - x_{ik})^2]^S$, where x_i, x_j are two expression profiles for genes i,*j* , *k* is the index of experiment (field), x_{ik} is the expression value of gene *i* in the experiment *k*.

(2) *Squared Euclidean distance.* The squared Euclidean distance can be implemented in order to place progressively greater weight on objects that are further apart. The squared Euclidian distance is computed as: $d_{ij} = \Sigma_k (x_{ik} - x_{ik})^2$ (see explanation above). The Euclidian and squared Euclidian distances are computed from raw data (non-standardized), therefore they may be affected by differences in scale among the expression values in different experiments.

(3) *Manhattan distance.* This distance is the average absolute difference for the set of experiments calculated by the formula $d_{ij} = \Sigma_k |x_{ik} - x_{ik}|$. In most cases, this distance measure yields results similar to the simple Euclidean distance, for this measure, the effect of single large differences is dampened (since they are not squared).

(4) *Chebychev distance.* This distance is computed as $d_{ij} = \max_k | x_{ik} - x_{ik} |$. The measure is useful when one wants to define two objects as "different" if they are different on any one of the experiments.

In SelTag all distance measures (1-3) are normalized to the number of fields involved in calculation. This is useful when take into account expression data with missing values.

Other measures involve correlation coefficient r_{ij} between two expression profiles of genes *i* and *j*.

(5) 1-r_{ij}; This measure keep close profiles with positive correlation coefficients and is useful when one wants to detect co-regulated genes.

(6) 1-$|r_{ij}|$; This measure keep close profiles with higher absolute value of correlation coefficients.

(7) 1+r_{ij}; This measure keep close profiles with negative value of correlation coefficients (anti-correlated).

Three types of correlation are possible for correlation distance option:
Pearson's *r* - Pearson's correlation coefficient. The Pearson product moment correlation coefficient between expression profiles *i* and *j* is calculated as follows:

$$r_{ij} = \frac{\sum_k (y_{ki} - \bar{y}_i)(y_{kj} - \bar{y}_j)}{(\sum_k (y_{ki} - \bar{y}_i)^2 \sum_k (y_{kj} - \bar{y}_j)^2)^{1/2}} ,$$

where y_{ki} is the expression level of gene *i* in the experiment *k*; $\bar{y}_i$ is the mean expression level of the gene *i*. Positive correlation implies that the expression levels of genes *i,j* are related positively, the higher expression of gene *i*, the higher expression of gene *j*. Negative correlation means that the expression levels of genes *i,j* are related negatively, the higher expression of gene *i*, the lower expression of gene *j*. If the r_{ij} is close to zero, two expression profiles are unrelated.
Spearman *r* - Spearman's correlation coefficient.
This correlation coefficient is computed for ranks. Let R_{ki} is the rank of the expression level in the experiment *k* of gene *i* (relatively to other experiments), R_{kj} is the rank of the expression level in the experiment *k* of gene *j*. Then Spearman's correlation coefficient is calculated by the formula

$$r_{ij}=\frac{\sum_k (R_{ki}-\bar{R}_i)(R_{kj}-\bar{R}_j)}{(\sum_k (R_{ki}-\bar{R}_i)^2 \sum_k (y_{kj}-\bar{R}_j)^2)^{1/2}}$$

<u>Kendall's τ</u> - Kendall's *tau* correlation coefficient.
To calculate Kendall's τ K for data points (y_{ki}; y_{kj}) 2K(K − 1) pairs considered (without self-pairing, the points in either order count as one pair). Pairs in which $y_{ki} > y_{mi}$ and $y_{kj} > y_{mj}$ or $y_{ki} < y_{mi}$ and $y_{kj} < y_{mj}$ are called concordant pairs (agreement between ranks), pairs with rank disagreement are called discordant pairs. In general, τ is calculated as

τ = ([number of concordant] – [number of discordant]) / total number of pairs

Clustering algorithm

The program performs nearest-neighbor clustering. If two expression profiles have distance lower than user-defined threshold, they form one cluster. If profile has distance lower than threshold to at least one profile from the cluster, it is added to the cluster.

When the cluster is defined, cluster scores are computed, that is average distance within the cluster. Gene score is the average distance from gene to other genes in the cluster (if size of cluster is greater than 1).

Example of the output data

```
status=Hierarchical clustering for cards...
status=9 clusters;Size:Min=1;Max=22.Get scores.
status=done [0.0 sec]
Number of clusters obtained 9.
Cluster Sizes and Scores:
Cluster 1       22      19044.5334
Cluster 2       3       5310.2424
Cluster 3       1       0.0000
Cluster 4       1       0.0000
Cluster 5       1       0.0000
Cluster 6       1       0.0000
Cluster 7       1       0.0000
Cluster 8       3       11528.7321
Cluster 9       1       0.0000
List of selected genes, their cluster indices and scores :
No      DataIndex       Name    Cluster Score
1       22      GEN20490        1       17400.0325
2       23      GEN35753        2       4479.8077
3       24      GEN02374        1       19743.1634
4       25      GEN32178        1       18608.6733
5       26      GEN06647        1       18895.3991
6       27      GEN34153        1       19301.8182
7       28      GEN00981        1       17364.7667
8       29      GEN07981        1       17494.5755
9       30      GEN20756        1       17584.5975
```

Some lines starting from "`status=`" are just output the status of the calculation and can be ignored. Then the result cluster information is output: number of clusters, their list with cluster scores. Then list of selected genes with their cluster indices and scores is printed out.

Parameter description:

Input	
Expression data	Input file in seltag format
Fields select	**List of fields** - List of expression fields (tissues) used to calculate correlation between gene expression profiles, namely field indices in data format of input file starting from 1 (column numeration is not depend on the Case Names option).

	Examples of input:
	1;2;3-7;12;
	1-12;
	Selection data - Filename for fields selection in XML format. This is another way to set the list of fields.

Genes for select	**List of genes** - List of genes to calculate correlation, namely gene indices in data set of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input:
	1;2;3-7;12;
	1-12;
	Gene list - Filename for genes selection in XML format for Gene List 1. This is another way to set the list of genes.

Output	
Result	Name of output file
Options	
Type of distance	Three types of distance are possible with respect to correlation coefficient rij: 1-rij; 1-|rij|; 1+rij
Type of correlation	Type of correlation coefficient. Three types of correlations are possible: Pearson's r, Spearman rank correlation and Kendall *tau* correlation.
Clustering threshold value	The value of clustering threshold
Missing data treatment	Option to treat missing data. Several options are possible : Substitute by means (missing data are substituted by expression means in corresponding field); Case-wise deletion (correlations/distances are calculated by excluding cases that have missing data for any of the selected variables, all correlations are based on the same set of data); Pair-wise deletion (correlations/distances between each pair of profiles are calculated from all fields/samples having valid data for those two profiles).

MAS5Baseline

Comparison of the Affymetrix gene expression row data to the baseline data by MAS 5.0 algorithm.

Data specification

The input for MAS5Baseline is the set of expression row data in Affymetrix CEL data format, corresponding CDF file and file with list of CEL files to be processed and their short description (this file is provided by user). The CEL file stores the results of the intensity calculations on the pixel values on the chip. The CDF file describes the layout for an Affymetrix GeneChip array. The output is SelTag data file with gene expression data. The baseline experiment name should be provided by user.

Algorithm description

The purpose of the algorithm is to perform noise correction and data normalization for each experiment and to estimate the change of the gene expression signal relatively to the baseline experiment signal. The method is known as MAS 5.0 statistical algorithm implemented in the Affymetrix Microarray Suite version 5.0. The algorithm details are described in the Affymetrix documentation at http://www.affymetrix.com/support/technical/technotesmain.affx ("Statistical

Algorithms Description Document", Affymetrix, 2002; "Statistical Algorithms Reference Guide", Affymetrix, 2001).

The algorithm contains of several steps.

1. Background noise correction for baseline and experiment
2. Change of the expression value (signal change) calculation between experiment and baseline
3. Estimation of the signal change value statistical significance (change detection p-values)
4. Estimation of the of the signal change (change detection call)

Background noise correction. At the first step the chip area is divided into K squared zones of the same size (default number of zones is 16). Then the 2% probes with the lowest intensity define the background intensity for each zone. The background noise level for each *k*-th zone bZ^k is the calculated as the average for those lowest intensity probes. The background noise level *b(x,y)* for each probe at the chip location *x,y* is calculated as weighted sum of zone background values

$$b(x,y) = \frac{1}{\sum_{k=1}^{K} w_k(x,y)} \sum_{k=1}^{K} w_k(x,y) bZ_k$$

,

where weights wk(x,y) are calculated as follows:

$$w_k(x,y) = \frac{1}{d_k^2(x,y) + smooth}$$

,

where $d_k(x,y)$ is the distance from the point *x,y* to the center of the *k*-th zone, *smooth* - is the smoothing parameter (by default is 100).

The noise correction procedure is as follows. First, standard deviations of the 2% probes with the lowest intensity nZ_k are calculated for each zone. For each probe the noise intensity *n(x,y)* is is estimated by above formulas (substitute *n(x,y)* for *b(x,y)* and nZ_k for bZ_k in the formulas above). Then the probe intensity corrected for noise is calculated from actual probe intensity *I(x,y)* as follows:

$A(x,y)=\max(I'(x,y)-b(x,y),NoiseFrac^{*}n(x,y))$,

where $I'(x,y)=\max(I(x,y),0.5)$, *NoiseFrac* is the fraction of noise and is set to 0.5 as in MAS 5.0 algorithm description.

Expression value (signal) calculation. After background subtraction from each probe intensity value, the signal values for the probesets are calculated. The calculation uses "ideal mismatch" technique that allows to process probe pairs for which the mismatch (MM) signal is greater than the match (PM) signal (see details in the Affymetrix documentation). When the ideal mismatch is calculated for each probe pair *j* of the each probeset *i*, the probe value PV_{ij} is calculated: $PV_{ij} = \log_2(\max(PM_{ij}-IM_{ij}, 2^{-20}))$. The signal log value ($SLV_i$) for the probeset *i* is calculated as the one-step biweight estimate for the corresponding probeset SLVs. Then the algorithm scales all the probesets to target scale value *Sc* (default is 500) estimating the scale factor *sf*

$$sf = \frac{Sc}{TrimMean\left(2^{SignalLogValue_i}, 0.02, 0.98\right)}$$

and using normalization factor *nf*:

$$nf = \frac{TrimMean\left(SPVb_{i,}\ 0.02, 0.98\right)}{TrimMean\left(SPVe_{i,}\ 0.02, 0.98\right)}$$

where $SPVb_i$ is the baseline signal, $SPVe_i$ is the experiment signal, the scaled probe intensity values are calculated as $SPV_{ij}=PV_{ij}+\ log_2(nf+sf)$. The *TrimMean* function calculates the mean value of the data without highest 2% and lowest 2% values. The probe log ratio *PLR* is calculated for probe pair *j* in probeset *i* on both the baseline *b* and experiment *e* arrays $PLR_{ij}={}_eSPV_{ij}-{}_bSPV_{ij}$. Having the probe log ratios *PLR* the *SignalLogRatio* is calculated using the biweight algorithm. *SignalLogRatio* is the reported value for this algorithm.

Estimation of the signal statistical significance (detection p-values). To estimate the significance of the change of the expression signal between experiment and baseline two additional sets of values for each probeset are calculated:

$$q_i = PM_i - MM_i,\ (i = 1, \ldots, n)$$

and

$$q_i = PM_i - MM_i,\ (i = 1, \ldots, n)$$

They are used to estimate two balancing factors:

$$nf = \frac{sfE}{sfB}$$

as the ratio of scaling factors of the of the *q* values for experiment *sfE* and baseline sfB data. The second balancing factor

$$nf_2 = \frac{sf_2E}{sf_2B}$$

is calculated as the ratio of scaling factors of the of the *z* values for experiment sf_2E and baseline sf_2B data. The balancing factor range is extended by using three balancing factors for the *q* values

$$f[0] = nf * d \qquad f[1] = nf \qquad f[2] = \frac{nf}{d}$$

and for *z* values

$$z_i = PM_i - b_i,\ (i = 1, \ldots, n)$$

where *d* is perturbation parameter and is set by default to 1.1.

If the algorithm settings indicate a user defined balancing factor and the factor is not equal to 1 then, *nf = nf2 = user defined normalization factor·sfE /sfB*, where *sfE* is the experiment *sf* and *sfB* is the baseline *sf* as described in the ***Expression value (signal) calculation*** section.

The critical *p*-value is estimated for all three *f*[*k*] (*k*=0,1,2) parameters and are designated below as *p*[0],*p*[1],*p*[2] correspondingly. These values are used to estimate the signal *p*-value for the signal change:

p=max(p[0],p[1],p[2]) if $p[0] < 0.5$, $p[1] < 0.5$ and $p[2] < 0.5$
p=min(p[0],p[1],p[2]) if $p[0] > 0.5$, $p[1] > 0.5$ and $p[2] > 0.5$
p=0.5 otherwise.

Estimation of the presence/absence of the signal (detection call). The algorithm report several types of detection calls in the output file: increase (I - is the designation of the detection call in the SelTag file), marginally increase but not increase (i), decrease (D), marginally decrease but not decrease (d), no change / unchanged (U). The definition of the detection change is dependent on several parameters: γ_1High, γ_1Low, γ_2High, γ_2Low, yielding two parameters γ_1 as linear interpolation of γ_1High and γ_1Low (if γ_1High = γ_1Low, then γ_1= γ_1High = γ_1Low), and 2 as linear interpolation of γ_2High and γ_2Low (if γ_2High = γ_2Low, then γ_2= γ_2High = γ_2Low).

The rule for the detection change is as follows:

$$\text{increase} \qquad \begin{cases} p[0] < \gamma_1 \\ p[1] < \gamma_1 \\ p[2] < \gamma_1 \end{cases}$$

$$\begin{array}{l}\text{marginally increase} \\ \text{but not increase}\end{array} \qquad \begin{cases} p[0] < \gamma_2 \\ p[1] < \gamma_2 \\ p[2] < \gamma_2 \end{cases}$$

$$\text{decrease} \qquad \begin{cases} p[0] > 1 - \gamma_1 \\ p[1] > 1 - \gamma_1 \\ p[2] > 1 - \gamma_1 \end{cases}$$

$$\begin{array}{l}\text{marginally decrease} \\ \text{but not decrease}\end{array} \qquad \begin{cases} p[0] > 1 - \gamma_2 \\ p[1] > 1 - \gamma_2 \\ p[2] > 1 - \gamma_2 \end{cases}$$

The MAS 5.0 default values for the gamma parameters are: γ_1High=0.0025, γ_1Low=0.0025; γ_2High=0.003, γ_2Low=0.003 (for 16-20 probe pairs).

Example of experiment list file

```
GSM42890        DEHP_48hr_Veh1  DEHP 48hr Veh1
GSM42891        DEHP_48hr_Veh2  DEHP 48hr Veh2
GSM42892        DEHP_48hr_Veh3  DEHP 48hr Veh3
GSM42893        DEHP_48hr_Veh4  DEHP 48hr Veh4
GSM42894        DEHP_48hr_Veh5  DEHP 48hr Veh5
```

This file contains three columns separated by symbol. First column is the experiment data name (the corresponding CEL file should start from this name and have extension *.cel, for example GSM42890.cel). Second column is the name of the variable in the output SelTag file, corresponding to this experiment (see below example of SelTag output file). This column should not contain spaces. Third column is the extended description of the experiment that will appear at the SelTag file header section.

Example of output data

```
#HEADER
```

```
Multiple chip data analysis by Affymetrix MAS5.0 algoritm [comparison with baseline].
ChipName=RG_U34A.
    BaselineDataFilename=GSM42895.cel.cel
    BaselineDataHeader=Baseline experiment
    BaselineDataScalingFactor=3.0104
    BaselineDataNormalizationFactor=1.0000
    BaselineDataSignalTrimmedMean=500.0000

  1 ExperimentDataFilename=GSM42907.cel
  1 DataHeader=VPA_48hr_Ve        VPA 48hr Veh POOLED
  1 DataScalingFactor=2.3930
  1 DataNormalizationFactor=1.0000
  1 DataSignalTrimmedMean=500.0000

  2 ExperimentDataFilename=GSM42913.cel
  2 DataHeader=DEHP_48hr_t        DEHP 48hr treated POOLED
  2 DataScalingFactor=2.6396
  2 DataNormalizationFactor=1.0000
  2 DataSignalTrimmedMean=500.0000

MAS5 algorithm parameters:
BF=2.0000
NZ=2.0000
Bsmooth=100.0000
Alpha1=0.0400
Alpha2=0.0600
Gamma1H=0.0025
Gamma1L=0.0025
Gamma2H=0.0030
Gamma2L=0.0030
Perturbation=1.1000
Tau=0.0150
TGT=500.0000
#ENDHEADER
ProbesetName     STRING
VPA_48hr_Ve_SignalLogRatio        FVALUE
VPA_48hr_Ve_Change       WORD
VPA_48hr_Ve_Change_p     FVALUE
DEHP_48hr_t_SignalLogRatio        FVALUE
DEHP_48hr_t_Change       WORD
DEHP_48hr_t_Change_p     FVALUE
END
DATA
AFFX-MurIL2_at   -0.0952 U        0.32868 -0.3230 U        0.28164
AFFX-MurIL10_at  0.5692  U        0.12112 0.3852  U        0.66645
AFFX-MurIL4_at   -0.1952 U        0.16996 -0.3095 U        0.30476
AFFX-MurFAS_at   -1.3517 U        0.49464 -0.2080 U        0.04914
AFFX-BioB-5_at   -0.7911 D        0.99998 0.0126  U        0.79768
AFFX-BioB-M_at   -0.7021 D        1.00000 -0.2708 D        0.99997
AFFX-BioB-3_at   -0.5249 D        0.99998 -0.4171 D        0.99987
```

Parameter description:

Input	
CDF file	The name of the CDF file for experiment set.
CEL directory	The name of the directory where all *.cel files can be found.
Experiment list file	File with experiment list and their description included into calculation.
Baseline experiment	Baseline experiment index.
Output	
Result	File with the resulting gene expression data in SelTag format.
Options	
Signal Only	If this flag set on, only signal values will be at the output. Otherwise,

	detection and detection *p*-values will be reported also.
Background floor	The percent of lowest intencity probes to be considered as background (MAS 5.0 default=2).
Zone number	Number of zones (K parameter) in background noise estimation. Default value for MAS 5.0 is 16.
Background smooth	The background weight smooth parameter (MAS 5.0 default=100).
Target signal	Target value for signal scaling (MAS 5.0 default =500).
Normalization factor	Normalization factor (default=1, i.e. the normalization factor is determined automatically).
Gamma1Low	Gamma1Low Parameter (MAS5.0 default is equal to Gamma1High = 0.0025).
Gamma1High	Gamma1High Parameter (MAS5.0 default is equal to Gamma1Low = 0.0025).
Gamma2Low	Gamma2Low Parameter (MAS5.0 default is equal to Gamma2High = 0.003).
Gamma2High	Gamma2High Parameter (MAS5.0 default is equal to Gamma2Low = 0.003).

MAS5Norm

Normalization of the Affymetrix gene expression row data by MAS 5.0 algorithm.

Data specification

The input for **MAS5Norm** is the set of expression row data in Affymetrix CEL data format, corresponding CDF file and file with list of CEL files to be processed ant their short description (this file is provided by user). The CEL file stores the results of the intensity calculations on the pixel values on the chip. The CDF file describes the layout for an Affymetrix GeneChip array. The output is SelTag data file with gene expression data.

Algorithm description

The purpose of the algorithm is to subtract background noise from the row probe intensities on the chip and perform data normalization to obtain normalized and scaled signal values for gene expression. The method is known as MAS 5.0 statistical algorithm implemented in the Affymetrix Microarray Suite version 5.0. The algorithm details are described in the Affymetrix documentation at http://www.affymetrix.com/support/technical/technotesmain.affx ("Statistical Algorithms Description Document", Affymetrix, 2002; "Statistical Algorithms Reference Guide", Affymetrix, 2001).

The algorithm contains of several steps.

1. Background noise correction
2. Expression value (signal) calculation
3. Estimation of the signal statistical significance (detection p-values)
4. Estimation of the presence/absence of the signal (detection call)

The algorithm contains of several steps.

1. Background noise correction for baseline and experiment
2. Change of the expression value (signal change) calculation between experiment and baseline

3. Estimation of the signal change value statistical significance (change detection p-values)
4. Estimation of the of the signal change (change detection call)

Background noise correction. At the first step the chip area is divided into K squared zones of the same size (default number of zones is 16). Then the 2% probes with the lowest intensity define the background intensity for each zone. The background noise level for each *k*-th zone bZ^k is the calculated as the average for those lowest intensity probes. The background noise level *b(x,y)* for each probe at the chip location *x,y* is calculated as weighted sum of zone background values

$$b(x,y) = \frac{1}{\sum_{k=1}^{K} w_k(x,y)} \sum_{k=1}^{K} w_k(x,y) bZ_k$$

,

where weights wk(x,y) are calculated as follows:

$$w_k(x,y) = \frac{1}{d_k^2(x,y) + smooth}$$

,

where $d_k(x,y)$ is the distance from the point *x,y* to the center of the *k*-th zone, *smooth* - is the smoothing parameter (by default is 100).

The noise correction procedure is as follows. First, standard deviations of the 2% probes with the lowest intensity nZ_k are calculated for each zone. For each probe the noise intensity *n(x,y)* is is estimated by above formulas (substitute *n(x,y)* for *b(x,y)* and nZ_k for bZ_k in the formulas above). Then the probe intensity corrected for noise is calculated from actual probe intensity *I(x,y)* as follows:

$A(x,y)=\max(I'(x,y)-b(x,y),NoiseFrac^{*}n(x,y))$,

where $I'(x,y)=\max(I(x,y),0.5)$, *NoiseFrac* is the fraction of noise and is set to 0.5 as in MAS 5.0 algorithm description.

Expression value (signal) calculation. After background subtraction from each probe intensity value, the signal values for the probesets are calculated. The calculation uses "ideal mismatch" technique that allows to process probe pairs for which the mismatch (MM) signal is greater than the match (PM) signal (see details in the Affymetrix documentation). When the ideal mismatch is calculated for each probe pair *j* of the each probeset *i*, the probe value PV_{ij} is calculated: $PV_{ij} = \log_2(\max(PM_{ij}-IM_{ij}, 2^{-20}))$. The signal log value ($SLV_i$) for the probeset *i* is calculated as the one-step biweight estimate for the corresponding probeset SLVs. Then the algorithm scales all the probesets to target scale value *Sc* (default is 500) estimating the scale factor *sf*

$$sf = \frac{Sc}{TrimMean\left(2^{SignalLogValue_i}, 0.02, 0.98\right)}$$

and using normalization factor *nf* (for this program is always set to 1):

$Signal=sf \cdot nf \cdot 2^{SLV}{}_i$. The *TrimMean* function calculates t he mean value of the data without highest 2% and lowest 2% values.

Estimation of the signal statistical significance (detection p-values). To estimate the significance of the signal deviation from noise Wilcoxon's rank test is used. This test determines the significance of the deviation of the discrimination score R_i for the probeset i

$$R_i = \frac{PM_i - MM_i}{PM_i + MM_i}$$

from the threshold value τ (this value specified by user, by default is set to 0.015). The significance of the deviation of the R_i from τ is calculated by Wilcoxon's rank test and reported as detection *p*-value.

Estimation of the presence/absence of the signal (detection call). The algorithm report three types of detection calls: present (P), marginal detection (M) or absent (A). The detection is based on the *p*-value and two user-defined parameters, α_1 and α_2: the signal is present if $p< \alpha_1$; the signal is marginally present if $\alpha_1 \leq p < \alpha_2$. The signal is absent if $p \leq 2$. By default $\alpha_1 = 0.04$ and $\alpha_2 = 0.06$ (for 16-20 probe pairs).

The program can analyze a set of CEL data files corresponding for the same CDF chip data. The output file is in SelTag format and reports the #HEADER section: Chip name; for each experiment (CEL file) ExperimentDataFilename, DataHeader as reported in the user-defined CEL list file, DataScalingFactor (*sf* value), DataNormalizationFactor (*nf* value), DataSignalTrimmedMean.

Example of experiment list file

```
GSM42890        DEHP_48hr_Veh1  DEHP 48hr Veh1
GSM42891        DEHP_48hr_Veh2  DEHP 48hr Veh2
GSM42892        DEHP_48hr_Veh3  DEHP 48hr Veh3
GSM42893        DEHP_48hr_Veh4  DEHP 48hr Veh4
GSM42894        DEHP_48hr_Veh5  DEHP 48hr Veh5
```

This file contains three columns separated by symbol. First column is the experiment data name (the corresponding CEL file should start from this name and have extension *.cel, for example GSM42890.cel). Second column is the name of the variable in the output SelTag file, corresponding to this experiment (see below example of SelTag output file). This column should not contain spaces. Third column is the extended description of the experiment that will appear at the SelTag file header section.

Example of output data

```
#HEADER
Multiple chip data analysis by Affymetrix MAS5.0 algoritm.
ChipName=RG_U34A.
  1 ExperimentDataFilename=GSM42890.cel
  1 DataHeader=DEHP_48hr_Veh1  DEHP 48hr Veh1
  1 DataScalingFactor=7.4530
  1 DataNormalizationFactor=1.0000
  1 DataSignalTrimmedMean=1500.0000
MAS5 algorithm parameters:
BF=2.0000
NZ=16
Bsmooth=100.0000
Alpha1=0.0400
Alpha2=0.0600
```

```
TGT=1500.0000
#ENDHEADER
ProbesetName    STRING
DEHP_48hr_Veh1_Signal  FVALUE
DEHP_48hr_Veh1_Detection       WORD
DEHP_48hr_Veh1_Detection_p     FVALUE
END
DATA
AFFX-MurIL2_at  37.5396 A       0.78955
AFFX-MurIL10_at 51.8929 A       0.60308
AFFX-MurIL4_at  5.7568  A       0.97607
AFFX-MurFAS_at  32.2922 A       0.60308
AFFX-BioB-5_at  714.0201        A       0.08359
AFFX-BioB-M_at  1563.2017       P       0.00125
AFFX-BioB-3_at  800.5414        P       0.00359
AFFX-BioC-5_at  3686.6155       P       0.00017
AFFX-BioC-3_at  1989.3492       P       0.00006
AFFX-BioDn-5_at 2807.6296       P       0.00066
AFFX-BioDn-3_at 16410.8984      P       0.00020
AFFX-CreX-5_at  32975.3750      P       0.00004
```

Parameter description:

Input	
CDF file	The name of the CDF file for experiment set.
CEL directory	The name of the directory where all *.cel files can be found.
Experiment list file	File with experiment list and their description included into calculation.
Output	
Result	File with the resulting gene expression data in SelTag format.
Options	
Signal Only	If this flag set on, only signal values will be at the output. Otherwise, detection and detection *p*-values will be reported also.
Alpha 1	Alpha 1 parameter for MAS 5.0 algorithm (MAS5.0 default is 0.04).
Alpha 2	Alpha 2 parameter for MAS 5.0 algorithm (MAS5.0 default is 0.06).
Background floor	The percent of lowest intencity probes to be considered as background (MAS 5.0 default=2).
Zone number	Number of zones (K parameter) in background noise estimation. Default value for MAS 5.0 is 16.
Background smooth	The background weight smooth parameter (MAS 5.0 default=100).
Target signal	Target value for signal scaling (MAS 5.0 default =500).
Tau	Tau parameter (MAS5.0 default is 0.015)

SelByExpr

Gene selection by query (logical expression).

Expression syntax

The logical expression contains field (experiment) indices denoted as $FX (where X is the field index) and relationships between values of the fields. For example, string

$F24 < 100

means that genes should be selected that have expression level for the field 24 lower then 100. To compare field values several operations can be used:

==	equal
<	less than

<=	less or equal to
>	greater than
>=	greater or equal to
!=	not equal

Complex queries may be formed using logical operations AND (&), OR (|), NOT (!) and parentheses for simple queries. For example, query

($F10 < 100) & ($F23 >= 0)

should return all genes with expression level in the experiment #10 lower than 100 and expression level in experiment #23 greater or equal to zero.

Some additional operations may be used also.

+,-	sum and difference
*,/	multiply and divide by
ABS(x)	absolute deviation of x
x^y	x in y power
SQRT(x)	square root of x

For example,

ABS($F10-$F11) < 100

Will select genes for which absolute deviation between expression levels in 10 and 11 experiments is lower than 100. Arithmetical operations are allowed with the numerical fields only.

Text comparison is also possible if the compared field is of the STRING or WORD types. For example, to select query with name "Gene2356" in the field $F1, one can set query

$F1=="Gene2356"

Note that the textual values is better to put in quotation marks, this will allow to process even strings containing spaces and special characters (arithmetical or logical operations described above).

Genes can be also selected by their numbers in data file, for example, query

$N <= 400

returns all genes with indices from 1 to 400.

Genes can be selected by their expression level in the field (experiment) group. For example, to select genes with the expression level greater than 100 in any of the experiment from group 1, the following query is applicable:

$G1 > 100

Condition level can be applied to the group selection, namely, user can specify the number of fields from the group satisfying condition. To select genes for which at least in 10 experiments expression level is greater than 100, the previous query can be modified:

$G1:10 > 100

The condition can be specified in percents of group size:

$G1:50% > 100

The latter query allow to select genes in which at least 50% experiments from group 1 have expression level greater than 100.

The score can be ascribed to the gene upon query evaluation. For example if the query is $F3 > 100 and there are two genes satisfying this condition with $F3 expression levels 105 and 800, the gene with expression level 800 will have greater score.

Example of the output data

```
List of selected genes and their scores [12 total]:
No      Index   Name    Score
1       1       GEN30482        0.5167
2       2       GEN03437        0.7767
3       3       GEN03687        0.9467
4       4       GEN24649        0.9600
5       5       GEN09108        0.2333
6       6       GEN09514        0.9933
7       7       GEN24589        0.7067
8       8       GEN02291        1.0233
```

```
9       9       GEN24534        0.9300
10      10      GEN14489        0.8000
11      11      GEN33519        0.8000
12      13      GEN35755        0.8633
```

First line is the header. It contains number of selected genes in parentheses. Second line is the data descriptions, separated by tabulation: No – number of the gene, Index – index of the gene in the large data file; Name – gene name (to determine name field in the data by default program searches the field that is called 'Name' in the field list names); Score – query scores (the better gene fits query expression, the higher the score). Next lines list data for selected genes separated by tabulation.

Parameter description

Input	
Expression data	File should contain expression data in seltag format.

Genes for select	**Genes for select** - List of genes to calculate correlation, namely gene indices in data set of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input:
	1;2;3-7;12;
	1-12;
	Gene list - Filename for genes selection in XML format for Gene List 1. This is another way to set the list of genes.

Output	
Result	Name of output file
Gene selection file	Selected genes can be additionally saved in XML file to be used further by MQSelTag. This parameter specify the name of the output selection file.
Name	Name of output selection.
Comment	Commentary for the output selection.
Options	
Query expressiom	Query expression in text format.

SelCorr

The program select most correlated genes for specified gene set.

Algorithm

The ***SelTag:SelCorr*** program allows selecting genes which have expression profiles highly correlated to the profile of the user-defined gene(s).
User should provide list of fields to calculate correlation.
Three types of correlation are possible:
<u>Pearson's *r*</u> - Pearson's correlation coefficient. The Pearson product moment correlation coefficient between expression profiles *i* and *j* is calculated as follows:

$$r_{ij}=\frac{\sum_k (y_{ki}-\bar{y}_i)(y_{kj}-\bar{y}_j)}{(\sum_k (y_{ki}-\bar{y}_i)^2\sum_k (y_{kj}-\bar{y}_j)^2)^{1/2}} \quad ,$$

where y_{ki} is the expression level of gene *i* in the experiment *k*; $\bar{y}_i$ is the mean expression level of the gene *i*. Positive correlation implies that the expression levels of genes *i,j* are related positively, the higher expression of gene *i*, the higher expression of gene *j*. Negative correlation

means that the expression levels of genes *i,j* are related negatively, the higher expression of gene *i*, the lower expression of gene *j*. If the r_{ij} is close to zero, two expression profiles are unrelated.
Spearman *r* - Spearman's correlation coefficient.
This correlation coefficient is computed for ranks. Let R_{ki} is the rank of the expression level in the experiment *k* of gene *i* (relatively to other experiments), R_{kj} is the rank of the expression level in the experiment *k* of gene *j*. Then Spearman's correlation coefficient is calculated by the formula

$$r_{ij}=\frac{\sum_k (R_{ki}-\bar{R}_i)(R_{kj}-\bar{R}_j)}{\left(\sum_k (R_{ki}-\bar{R}_i)^2 \sum_k (y_{kj}-\bar{R}_j)^2\right)^{1/2}}$$

Kendall's τ - Kendall's *tau* correlation coefficient.
To calculate Kendall's τ K for data points (y_{ki}; y_{kj}) 2K(K − 1) pairs considered (without self-pairing, the points in either order count as one pair). Pairs in which $y_{ki} > y_{mi}$ and $y_{kj} > y_{mj}$ or $y_{ki} < y_{mi}$ and $y_{kj} < y_{mj}$ are called concordant pairs (agreement between ranks), pairs with rank disagreement are called discordant pairs. In general, τ is calculated as
τ = ([number of concordant] – [number of discordant]) / total number of pairs

For the specified gene user can select other genes that have correlation coefficient between target gene expression profile greater than threshold. There are several threshold types: "Best N" - select N most correlated genes from set; "Best %" - select a fraction (in %) of most correlated genes from set; "Value" - select the genes with the absolute correlation value equal or higher than the threshold; "All" - select all genes from list.
If a number of genes are selected in target list, several options exist how to treat the correlation of profile with this groups of profiles: "Max. correlation value to select" - when comparing genes, the key parameter is the maximum coefficient of correlation of a gene from Set 1 with genes from Set 2; "Aver. correlation value to select" - when comparing genes from Set 1, the key parameter is the average coefficient of the correlation of a gene from Set 1 with genes from Set 2; "Corr. for aver. field values to select" - when comparing genes from Set 1, the key parameter is the coefficient of correlation of a gene from Set 2 with an average profile of genes from Set 2. This means that the program creates an "imaginary" average gene from Set 2 and uses this average value to calculate the correlation coefficient.

Example of the output data

```
status=Correlation matrix for cards...
status=Correlation matrix calculation...
status=done [0.0 sec]
List of selected genes [30 total]:
1     6718  X54232
2     4575  R81175
3     7132  X79981
4     5493  T78432
5     3454  R06627
6     5166  T59895
7     6042  U14394
8     6690  X52947
```

Some lines starting from “`status=`″ just output the status of the calculation and can be ignored. Then the result information (with the number of selected genes) is output. Then list of selected genes with their indices in data file and gene names are printed out.

Parameter description

Input	
SelTag data	Input file in seltag format

Fields select	**List of fields** - List of expression fields (tissues) used to calculate correlation between gene expression profiles, namely field indices in data format of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input:
	1;2;3-7;12;
	1-12;
	Fields list - Filename for fields selection in XML format. This is another way to set the list of fields.

Genes for select	**Genes for select** - List of genes to calculate correlation, namely gene indices in data set of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input:
	1;2;3-7;12;
	1-12;
	Gene list - Filename for genes selection in XML format for Gene List 1. This is another way to set the list of genes.

Genes for comparison	**Genes for comparison** - List of genes to which calculate correlation, namely gene indices in data set of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input:
	1;2;3-7;12;
	1-12;
	Gene list - Filename for genes selection in XML format for Gene List 2. This is another way to set the list of genes.

Output	
Result	Name of output file
Correlation matrix	Output correlation matrix for selected genes
XML data	Name of the file for graphical output of correlation coefficient value profiles. If not specified then no graph output assumed.
Title	User-specified title of the graph plot.
Author	User-specified name of the graph author.
Comment	User-specified graph additional commentary line.
X axis name	User-specified graph X axis name.
Y axis name	User-specified graph Y axis name.
Gene selection file	Selected genes can be additionally saved in XML file to be used further by MQSelTag. This parameter specify the name of the output selection file.
Name	Name of output selection.
Comment	Commentary for the output selection.
Options	
Type of correlation	Type of correlation coefficient. Three types of correlations are possible: **Pearson's r**, **Spearman rank correlation** and **Kendall *tau*** correlation.
Selection regime	Regime to treat multiple genes to compare with single gene. Several options are possible: **Max. correlation value to select** - the maximal correlation value between expression profiles in gene set to query gene is evaluated; **Aver. correlation value to select** - average correlation coefficient value is calculated; **Corr. for aver field values to select** - mean expression values are calculated in the set of genes and their correlation for the query expression profile is calculated.

Correlation threshold type	Type of threshold to select best correlating gene pairs. Several options are possible: Best N correlations ; Best % correlations; Correlation coefficient value; Select all pairs.
Correlation threshold value	Threshold to select genes from List 1 on the basis of the their correlation coefficient value to genes from List 2.
Missing data treatment	Option to treat missing data. Several options are possible : Substitute by means (missing data are substituted by expression means in corresponding field); Case-wise deletion (correlations/distances are calculated by excluding cases that have missing data for any of the selected variables, all correlations are based on the same set of data); Pair-wise deletion (correlations/distances between each pair of profiles are calculated from all fields/samples having valid data for those two profiles).

SOMClust

Algorithm description

SOM (Self-organizing map) algorithm was suggested for unsupervised learning problems solution (i.e. classification) by Kohonen [Kohonen, T. (1997) Self-Organizing Maps (Springer, Berlin)]. The algorithm provides mapping from high-dimensional data to low-dimensional space (2D). The SOM clustering was used for expression data analysis by Tamayo *et al*. [Tamayo P. et al (1999) Proc. Natl. Acad. Sci. USA, 96, 2907–2912]. The approach of Tomayo *et al* is implemented in SelTag.

An SOM has a set of nodes with a simple topology (e.g., two-dimensional grid) and a distance function d(N1,N2) on the nodes. Nodes are mapped into *K*-dimensional ‘‘gene expression’’ space (in which the *i*-th coordinate represents the expression level in the *i*-th sample, *K* is the number of experiments (fields)). The process of mapping is iterative (see Fig.1).

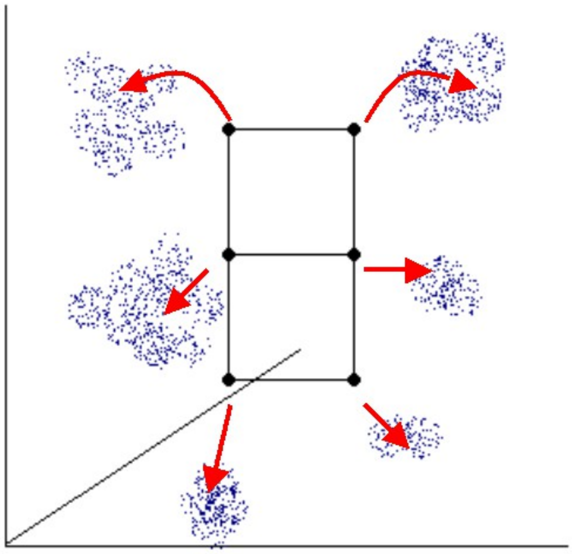


Fig. 1. The diagram shows the principle of iterative clustering of high-dimensional data points by SOM algorithm. The SOM structure is shown by black grid, data points in high-dimensional space are shown in blue. The moving of grid nodes to the regions of higher data density are shown in red.

The iterative algorithm allows moving each node to the *K*-dimensional space regions with higher density of points (genes). In principle, each node will be located near the cluster of genes in the high-dimensional space. The position of node N at iteration *i* is denoted $f_i(N)$. The initial mapping f_0 is random. On subsequent iterations, a data point P is selected and the node N_P that maps nearest to P is identified. The mapping of nodes is then adjusted by moving points toward P by the formula (Tomayo *et al,*1999):

$$f_{i+1}(N) = f_i(N) + \tau(d(N, N_P), i)\ (P - f_i(N)).$$

To perform calculation user should define the grid size (number of row and column nodes in two-dimensional grid (see Fig.1), set the maximal number of iterations and set the distance type (to calculate distance between node and data points). There are several measures of expression profile distance between two genes:

(1) *Euclidean distance.* This is the geometric distance in the multidimensional space. It is computed as: $d_{ij} = [\Sigma_k (x_{ik} - x_{ik})^2]^S$, where x_i, x_j are two expression profiles for genes i,*j* , *k* is the index of experiment (field), x_{ik} is the expression value of gene *i* in the experiment *k.*

(2) *Squared Euclidean distance.* The squared Euclidean distance can be implemented in order to place progressively greater weight on objects that are further apart. The squared Euclidian distance is computed as: $d_{ij} = \Sigma_k (x_{ik} - x_{ik})^2$ (see explanation above). The Euclidian and squared Euclidian distances are computed from raw data (non-standardized), therefore they may be affected by differences in scale among the expression values in different experiments.

(3) *Manhattan distance.* This distance is the average absolute difference for the set of experiments calculated by the formula $d_{ij} = \Sigma_k |x_{ik} - x_{ik}|$. In most cases, this distance measure yields results similar to the simple Euclidean distance, for this measure, the effect of single large differences is dampened (since they are not squared).

(4) *Chebychev distance.* This distance is computed as $d_{ij} = \max_k | x_{ik} - x_{ik} |$. The measure is useful when one wants to define two objects as "different" if they are different on any one of the experiments.

In SelTag all distance measures (1-3) are normalized to the number of fields involved in calculation. This is useful when take into account expression data with missing values.

Other measures involve correlation coefficient r_{ij} between two expression profiles of genes *i* and *j*.

(5) 1-r_{ij}; This measure keep close profiles with positive correlation coefficients and is useful when one wants to detect co-regulated genes.

(6) 1-|r_{ij}|; This measure keep close profiles with higher absolute value of correlation coefficients.

(7) 1+r_{ij}; This measure keep close profiles with negative value of correlation coefficients (anti-correlated).

Three types of correlation are possible for correlation distance option:
Pearson's *r* - Pearson's correlation coefficient. The Pearson product moment correlation coefficient between expression profiles *i* and *j* is calculated as follows:

$$r_{ij}=\frac{\sum_k (y_{ki}-\bar{y}_i)(y_{kj}-\bar{y}_j)}{(\sum_k (y_{ki}-\bar{y}_i)^2 \sum_k (y_{kj}-\bar{y}_j)^2)^{1/2}} \;,$$

where y_{ki} is the expression level of gene *i* in the experiment *k*; $\bar{y}_i$ is the mean expression level of the gene *i*. Positive correlation implies that the expression levels of genes *i,j* are related positively, the higher expression of gene *i*, the higher expression of gene *j*. Negative correlation means that the expression levels of genes *i,j* are related negatively, the higher expression of gene *i*, the lower expression of gene *j*. If the r_{ij} is close to zero, two expression profiles are unrelated.

<u>Spearman *r*</u> - Spearman's correlation coefficient.

This correlation coefficient is computed for ranks. Let R_{ki} is the rank of the expression level in the experiment *k* of gene *i* (relatively to other experiments), R_{kj} is the rank of the expression level in the experiment *k* of gene *j*. Then Spearman's correlation coefficient is calculated by the formula

$$r_{ij}=\frac{\sum_k (R_{ki}-\bar{R}_i)(R_{kj}-\bar{R}_j)}{(\sum_k (R_{ki}-\bar{R}_i)^2 \sum_k (y_{kj}-\bar{R}_j)^2)^{1/2}}$$

<u>Kendall's τ</u> - Kendall's *tau* correlation coefficient.

To calculate Kendall's τ K for data points (y_{ki}; y_{kj}) 2K(K − 1) pairs considered (without self-pairing, the points in either order count as one pair). Pairs in which $y_{ki} > y_{mi}$ and $y_{kj} > y_{mj}$ or $y_{ki} < y_{mi}$ and $y_{kj} < y_{mj}$ are called concordant pairs (agreement between ranks), pairs with rank disagreement are called discordant pairs. In general, τ is calculated as

τ = ([number of concordant] – [number of discordant]) / total number of pairs

Example of output data

```
status=done [0.0 sec]
Number of gene clusters obtained 4.
Cluster Sizes and Scores:
Cluster 1      2       1.1201
Cluster 2      5       0.5954
Cluster 3      19      0.8783
Cluster 4      8       0.7907
List of selected genes, their cluster indices and scores :
No      DataIndex       Name    Cluster Score
1       1       GEN30482        1       1.1201
2       2       GEN03437        1       1.1201
3       3       GEN03687        2       0.7264
```

Some lines starting from "`status=`" are just output the status of the calculation and can be ignored. Then the result cluster information is output: number of clusters, their list with cluster scores. Some clusters (grid nodes) may not contain any genes, they omitted from the output. Then list of selected genes with their cluster indices and scores is printed out.

Parameter description

Input	
SelTag data	Input file in seltag format
Fields select	**List of fields** - List of expression fields (tissues) used to calculate correlation between gene expression profiles, namely field indices in data format of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input: 1;2;3-7;12;

	1-12; **Fields list** - Filename for fields selection in XML format. This is another way to set the list of fields.

Genes for select	**Genes for select** - List of genes to calculate correlation, namely gene indices in data set of input file starting from 1 (column numeration is not depend on the Case Names option). Examples of input:
	1;2;3-7;12;
	1-12;
	Gene list - Filename for genes selection in XML format for Gene List. This is another way to set the list of genes.

Output	
Result	Name of output file
options	Number of rows in grid This parameter defines number of rows in the map
Number of columns in grid	This parameter defines number of columns in the map
Options	
Select clustering objects	Select clustering objects: genes or samples
Type of distance	Type of distance between expression profiles. Several types of correlations are possible: 1-r_{ij}; 1-|r_{ij}|; 1+r_{ij}; Squared Euclidian distance; Euclidian distance; Manhattan distance; Chebyshev distance.
Missing data treatment	Option to treat missing data. Several options are possible: **Substitute by means** (missing data are substituted by expression means in corresponding field); **Case-wise deletion** (correlations/distances are calculated by excluding cases that have missing data for any of the selected variables, all correlations are based on the same set of data); **Pair-wise deletion** (correlations/distances between each pair of profiles are calculated from all fields/samples having valid data for those two profiles).
Maximal number of iterations	Maximal number of iterations to perform SOM clustering.

Sequences Manipulation

AddSeq

Add the second sequence to end of the first sequence.

Parameters:

Input	
Target sequence	Name of the input file
Additional sequence	Name of the additional file
Output	
Result	Name of the output file

Complement

Generation of complementary DNA or RNA sequence.

Parameters:

Input	
Sequence	Name of the input file
Output	
Result	Name of the output file
Options	
Operation	Select sequence operation: **Complement** - create a complementary sequence (chain -). **Reverse** - make a reverse order sequence.

CutGet

Simple Cut/Get sequence.
CutGet serves to allocation of a fragment from a sequence or cutting out (deletion) of a fragment from a sequence.

Parameters:

Input	
Sequence	Name of the input file
Output	
Result	Name of the output file
Options	
Operation	Select sequence operation. Select sequence operation: **Cut** - remove the symbols from sequence position. **Get** - get part of sequence
Set Range	Set Range: **From** - Set the starting position for a fragment of sequence. **To** - Set the ending position for a fragment of sequence.

GetSeq

Extracts sequence from a file.

Parameters:

Input

Data	Name of the input file
Output	
Result	Name of the output file
String length	Count of symbols by line (default value is 60)
Options	
Type of sequence	Type of sequence: **DNA bases** - ATGC **RNA bases** - AUGC **DNA bases+N** - ATGCN (N - unknown) **RNA bases+N** - AUGCN (N - unknown) **Standard aminoacids** - AVLICMPYFWDNEQHSTKRG

InsSeq

Insert the second sequence to a specific position of the first sequence.
Parameters:

Input	
Target sequence	Name of the input file
Insert sequence	Name of the additional file
Output	
Result	Name of the output file
Options	
Position	Insert position

Motif

The program performs motif search on a sequence
This program use statistical functions from "R" free software environment for statistical computing and graphics (http://www.r-project.org). This program requires the R-package to be installed on your computer.
Parameters:

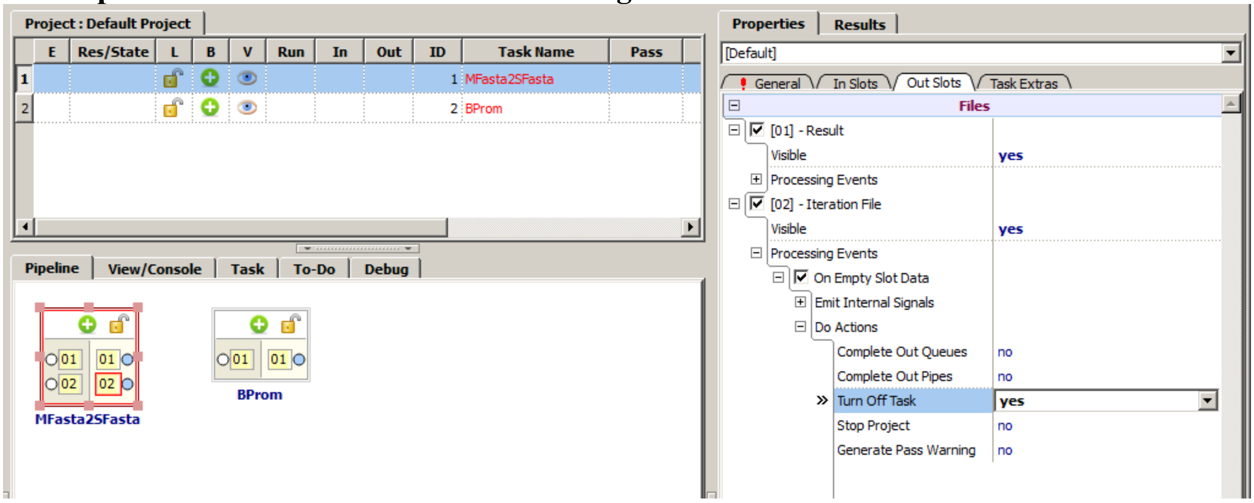
Input	
Nucleotide sequence	File with the sequence to search for (sequence must not contain gaps, otherwise the motif shall not be found).
Output	
Result	Name of the output file.
Strand to Search in	Scan sequence in selected strand(s). **In direct strand only** - Direct strand will be searched. **In reverse strand only** - Reverse strand will be searched. **In both strands** - Both direct and reverse strands will be searched.
String to Search for	The sequence to search for (sequence must not contain gaps, otherwise the motif shall not be found).
Mismatch Limit	Maximal percentage of mismatches.

MFasta2SFasta

MFasta2SFasta serves to split multifasta files into singlefasta ones.
Example of use MFasta2SFasta with BPROM
Current version of MolQuest does not support multifasta files for BPROM input. However, it is possible to process multifasta files via the following sequence of procedures: split multifasta files

into singlefasta ones with use of the MFasta2SFasta utility and then send them to BPROM via pipelines. This requires the following steps.
Add to project MFasta2SFasta and BPROM. It requires to open the "Toolbox>Programs" item and double click on appropriate program/utility. MFasta2SFasta can be found in the "All programs" and "SeqMan" list sections, BPROM – in the "All programs" and "Eukaryotic gene Finding" ones.

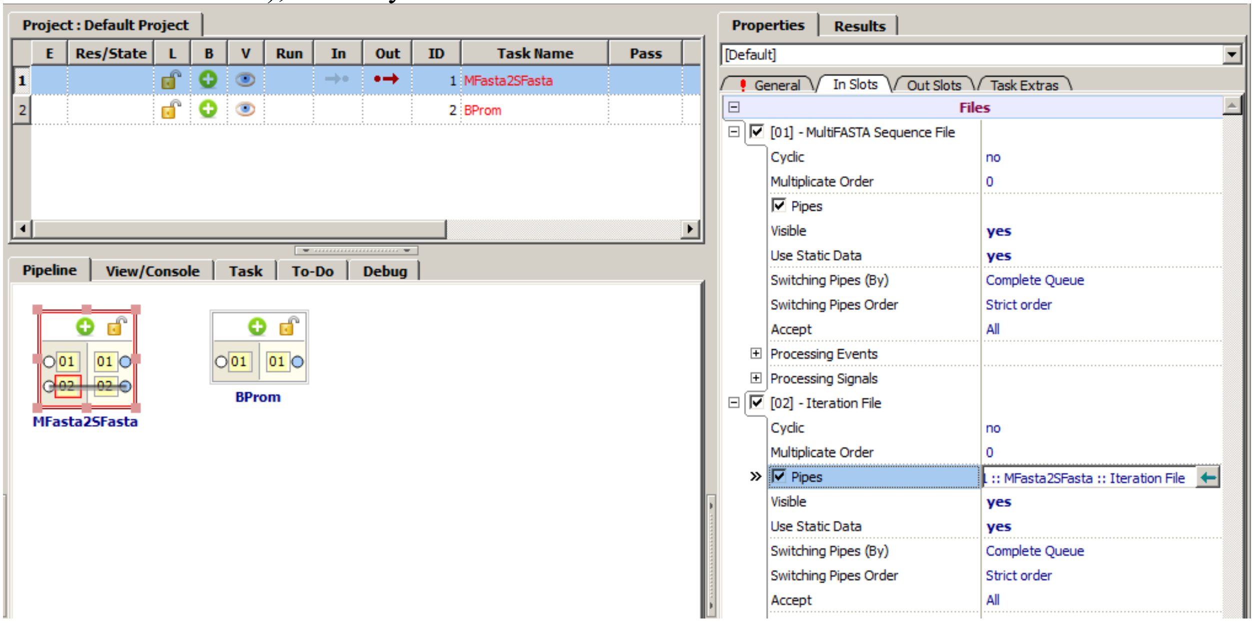
In setup of of MFasta2SFasta do the following:



	E	Res/State	L	B	V	Run	In	Out	ID	Task Name	Pass
1									1	MFasta2SFasta	
2									2	BProm	

☑ [01] - Result
Visible: yes
Processing Events
☑ [02] - Iteration File
Visible: yes
Processing Events
☑ On Empty Slot Data
Emit Internal Signals
Do Actions
Complete Out Queues: no
Complete Out Pipes: no
Turn Off Task: yes
Stop Project: no
Generate Pass Warning: no

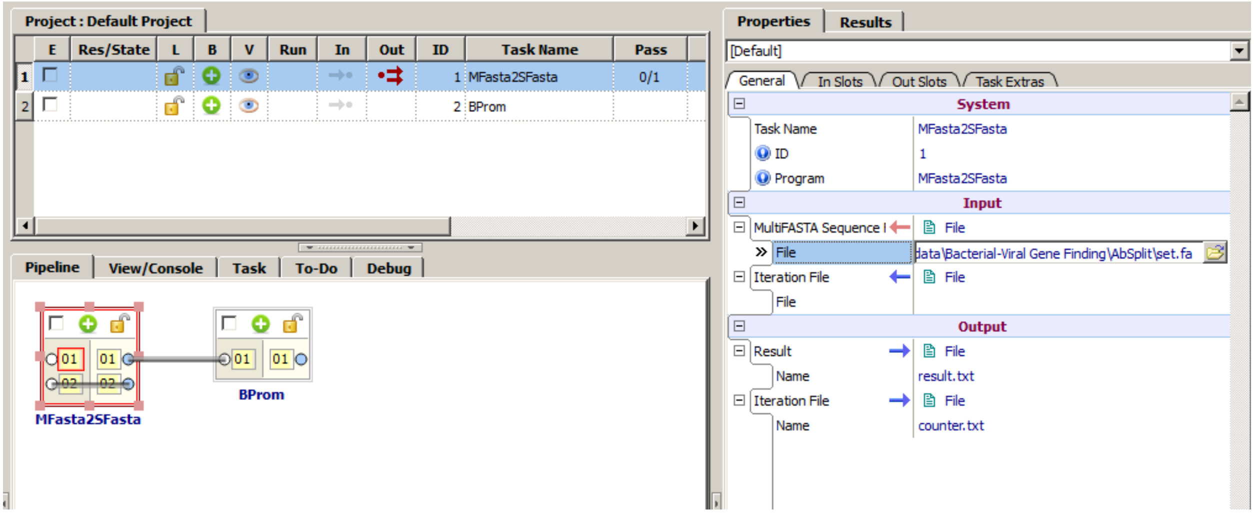
1. Find the "Iteration File" parameter (can be found in the "Properties" tab, in the "Out Slots" subtab), and in the "Processing Events" node check in the "On Empty Slot Data" check-box and further in the "Do Actions" subnode set the "Turn Off Task" value to "yes". 2. Link the output "[02] - Iteration File" to its input "[02] - Iteration File" (can be found in the "Properties" tab, in the "In Slots" subtab), i.e. to cycle it to itself.



	E	Res/State	L	B	V	Run	In	Out	ID	Task Name	Pass
1									1	MFasta2SFasta	
2									2	BProm	

☑ [01] - MultiFASTA Sequence File
Cyclic: no
Multiplicate Order: 0
☑ Pipes
Visible: yes
Use Static Data: yes
Switching Pipes (By): Complete Queue
Switching Pipes Order: Strict order
Accept: All
Processing Events
Processing Signals
☑ [02] - Iteration File
Cyclic: no
Multiplicate Order: 0
☑ Pipes: 1 :: MFasta2SFasta :: Iteration File
Visible: yes
Use Static Data: yes
Switching Pipes (By): Complete Queue
Switching Pipes Order: Strict order
Accept: All

For MFasta2SFasta do the following:
1. To the slot "MultiFASTA Sequence File" (can be found in the "Properties" tab, in the "General" subtab) send a file with sequences (multifasta).
2. The first out slot "1 :: MFasta2SFasta :: Result" of the MFasta2SFasta link with the in slot "Sequences set" ("01") of the BPROM.



Project : Default Project

	E	Res/State	L	B	V	Run	In	Out	ID	Task Name	Pass
1	□								1	MFasta2SFasta	0/1
2	□								2	BProm	

[Default]

System
Task Name MFasta2SFasta
ID 1
Program MFasta2SFasta

Input
MultiFASTA Sequence I File
File data\Bacterial-Viral Gene Finding\AbSplit\set.fa
Iteration File File
File

Output
Result File
Name result.txt
Iteration File File
Name counter.txt

Parameters

Input	
MultiFasta sequence file	File with sequences in multi-FASTA format.
Iteration file	Iteration file.
Output	
Result	Name of the output file.

OligoMap

Program for fast mapping a big set of oligos to chromosome sequences

OligoMap is designed to map a set of oligonucleotides used for microarray production. The program maps 300,000 25-30 bp long oligos on 49 MB of unmasked chromosome 22 in 8 min. Program is useful to check locations of oligos and their uniqueness in genome. Its output is similar to that of EstMap.

Output example

```
Sequence 1 found: 1
L:49396972   Sequence chr22
[DD] Sequence:       1(      1), S:           0, L:       22 cut1 of chr22
Block of alignment: 1
    1 P:   49014410         1 L:       22, G: 100.00, W:     220, S:7.77817

-----------------------------------------------------------------------

Sequence 2 found: 12
L:246127941  Sequence chr1
[DD] Sequence:       2(      1), S:           0, L:       18 cut2 of chr22
Block of alignment: 1
    1 P: 199136157         1 L:       18, G:  94.44, W:     150, S:6.45497
L:199344050  Sequence chr3
[DR] Sequence:       2(      1), S:           0, L:       18 cut2 of chr22
Block of alignment: 1
    1 P:  11683162         1 L:       18, G:  94.44, W:     150, S:6.45497
L:170914576  Sequence chr6
[DR] Sequence:       2(      1), S:           0, L:       18 cut2 of chr22
Block of alignment: 3
    1 P:   3133720         1 L:       18, G:  88.89, W:     120, S:5.93857
    2 P:  62375122         1 L:       18, G:  88.89, W:     120, S:5.93857
    3 P:  51740936         1 L:       18, G:  88.89, W:     120, S:5.93857
L:146308819  Sequence chr8
```

```
[DR] Sequence:       2(      1), S:           0, L:       18 cut2 of chr22
Block of alignment: 1
    1 P:  60080010         1 L:      18, G:  88.89, W:    120, S:5.93857
L:134482954  Sequence chr11
[DR] Sequence:       2(      1), S:           0, L:       18 cut2 of chr22
Block of alignment: 2
    1 P:  81210160         1 L:      18, G:  94.44, W:    150, S:6.45497
    2 P:  45434208         1 L:      18, G:  88.89, W:    120, S:5.93857
L:132078379  Sequence chr12
[DR] Sequence:       2(      1), S:           0, L:       18 cut2 of chr22
Block of alignment: 1
    1 P:  49358387         1 L:      18, G:  94.44, W:    150, S:6.45497
L:76115139   Sequence chr18
[DD] Sequence:       2(      1), S:           0, L:       18 cut2 of chr22
Block of alignment: 1
    1 P:  73733199         1 L:      18, G:  94.44, W:    150, S:6.45497
L:63811651   Sequence chr19
[DR] Sequence:       2(      1), S:           0, L:       18 cut2 of chr22
Block of alignment: 1
    1 P:  60444721         1 L:      18, G:  88.89, W:    120, S:5.93857
L:49396972   Sequence chr22
[DD] Sequence:       2(      1), S:           0, L:       18 cut2 of chr22
Block of alignment: 1
    1 P:  49014360         1 L:      18, G: 100.00, W:    180, S:6.97137

-----------------------------------------------------------------------
Sequence 3 found: 54
L:246127941  Sequence chr1
[DD] Sequence:       3(      1), S:           0, L:       16 cut3 of chr22
Block of alignment: 5
    1 P: 231124663         1 L:      16, G:  93.75, W:    130, S:5.98764
    2 P:  38695182         1 L:      16, G:  87.50, W:    100, S:5.44331
    3 P: 211588869         1 L:      16, G:  87.50, W:    100, S:5.44331
    4 P: 225236371         1 L:      16, G:  93.75, W:    130, S:5.98764
    5 P:    932675         1 L:      16, G:  87.50, W:    100, S:5.44331
[DR] Sequence:       3(      1), S:           0, L:       16 cut3 of chr22
Block of alignment: 1
    1 P:  39839150         1 L:      16, G:  87.50, W:    100, S:5.44331
L:243615958  Sequence chr2
[DR] Sequence:       3(      1), S:           0, L:       16 cut3 of chr22
Block of alignment: 1
    1 P: 157495379         1 L:      16, G:  87.50, W:    100, S:5.44331
L:199344050  Sequence chr3
[DR] Sequence:       3(      1), S:           0, L:       16 cut3 of chr22
Block of alignment: 1
    1 P:  52046346         1 L:      16, G:  93.75, W:    130, S:5.98764
L:191731959  Sequence chr4
[DR] Sequence:       3(      1), S:           0, L:       16 cut3 of chr22
Block of alignment: 1
    1 P: 137560710         1 L:      16, G:  87.50, W:    100, S:5.44331
L:181034922  Sequence chr5
[DD] Sequence:       3(      1), S:           0, L:       16 cut3 of chr22
Block of alignment: 1
    1 P:  74433239         1 L:      16, G:  87.50, W:    100, S:5.44331
[DR] Sequence:       3(      1), S:           0, L:       16 cut3 of chr22
Block of alignment: 1
    1 P: 180126965         1 L:      16, G:  87.50, W:    100, S:5.44331
L:170914576  Sequence chr6
[DD] Sequence:       3(      1), S:           0, L:       16 cut3 of chr22
Block of alignment: 1
    1 P:  30136862         1 L:      16, G:  87.50, W:    100, S:5.44331
L:158545518  Sequence chr7
[DD] Sequence:       3(      1), S:           0, L:       16 cut3 of chr22
Block of alignment: 1
```

```
    1 P:   1168967         1 L:      16, G:  87.50, W:    100, S:5.44331
[DR] Sequence:       3(      1), S:          0, L:       16 cut3 of chr22
Block of alignment: 1
    1 P: 122887080         1 L:      16, G:  87.50, W:    100, S:5.44331
L:146308819  Sequence chr8
[DD] Sequence:       3(      1), S:          0, L:       16 cut3 of chr22
Block of alignment: 4
    1 P:   7403617         1 L:      16, G:  87.50, W:    100, S:5.44331
    2 P: 145427481         1 L:      16, G:  87.50, W:    100, S:5.44331
    3 P:  74709150         1 L:      16, G:  87.50, W:    100, S:5.44331
    4 P:  95309818         1 L:      16, G:  87.50, W:    100, S:5.44331
...
```

Oligs

The program makes statistical calculations on oligonucleotides (4-nucleotides) and shows the ones of significant differences to expected mean.

Input data

The input file should be in FASTA format and may contain several sequences. Alphabet. The allowed symbols: "ACGTUacgtu" and "NnyYrRBbDdHhKkWwSsMmVv". The symbols to be skipped: "0123456789; \n\r\t\0-". All other symbols are not allowed.

Input parameters

The program processes all oligonucleotides of length L. The L value runs all values in L1 to L2 range.

Minimal olig length (L1) - Minimal olig length

Minimal olig length (L2) - Minimal olig length

Restrictions for L1, L2: 1<=L1 && L1<=L2 && L2<=13.

Computer must have enough memory installed, and the memory size depends on oligo's length.

Input file - Input file in FASTA-format.

The special mode to print all oligos ignoring any additional conditions. While in this mode the very big output file can be generated.

Print all oligs - Print all oligs, ignore conditions

The program can process not only the given sequence but simultaneously build and process the reverse sequence.

Scan target sequence in different chain - Scan target sequence in different chain:
In direct chain only (default)
In reverse chain only
In both chains

Similarly to normal distribution, the program can output either most frequent oligos or most rare ones. The following parameter is used for this:

Frequency - Most frequent or least frequent:
most frequent (default)
least frequent

To determine which oligos must be output and which ones must not, the value for deviation multiplier range should be defined.

Deviation multiplier is difference between number of oligos and expected number of oligos in sigma units. For more details see the algorithm description chapter.

Deviation multiplier fence - Use the value 3.0 to output 5% of oligos.

Output file - Output file name.

The "shift" parameter sets the value (in nucleotides) of shifting from the sequence start to the position from which oligos are to be generated. If there are several sequences in a file, the shift value affects each of them. The default value is 0.

Shift in sequence - Shift in sequence, default value is 0.

The "step" parameter sets the value (in nucleotides) of shifting for generating oligos. In order to get all oligos, this parameter should be set to 1, which is default value.

Step in sequence - Step in sequence (default value is 1)

Sometime it's necessary to check all three reading frames. To do this run the program three times with the following values for "shift" and "step":

1) step=3 shift=0
2) step=3 shift=1
3) step=3 shift=2

Input sequences may be either in FASTA format or in specially packed format. The "Softberry" products frequently used to pack large chromosomes into its own "nucfile" or nf format. Sequence file, in this case, has the .nf extension.

If the "Packed file" parameter is not defined the program consider the input file as one in FASTA format. Otherwise the input file format is considered as "nucfile".

Packed file - Input file is packed file (nucfile, nf).

The FASTA file can be converted to the nucfile one using the cvtseq utility.
For example, to convert the FASTA file chr22.fa to the nucfile chr22.nf, use the following command string:

```
cvtseq chr22.fa chr22.nf -fi -do -t "chr22" -n5gc
```

Use the following command to check the information on a packed file:

```
cvtseq chr22.nf -e
```

Command output:

```
filename: chr22.nf
pack_mode: PACK_5

size: 49476972 from: 0 nonstandard: 1
title_size: 5 title: chr22
```

Algorithm

For each defined L the array that contains the number of oligos is built. The sequential number of oligo is used as an index for this array. The total number of oligos is a value of the array.

Further, using this array and defined parameters, program builds the table of oligos that contains more information (mean, deviation multiplier etc). This table is printed into output file.

Total number of all oligos - oligs_sum_count.
Total number of nucleotides - seqs_sum_length.

The oligo's frequency is a multiplication of frequencies of nucleotides it consists of.

The expected mean of the counter (that is equal to oligo's mean) is calculated by the following way:

average= oligs_sum_count*frequence;

Deviation is calculated with use of formula:

deviation = sqrt(oligs_sum_count*frequence*(1-frequence));

The oligo's counter - olig_count - describes how much times this oligo occurs in a sequence.

Deviation multiplier is calculated with use of formula:

Deviation_multiplier= (olig_count-average)/deviation;

Normalized deviation (norm deviate) of the given oligo is calculated with use of formula:

Norm_deviate= olig_count/seqs_sum_length;

Output data

```
Example for program output:

Oligs 1.6  Copyright (c) 2005-2006 Softberry
Num seqs=32 Nucleotides=46705 Average seq length=1459.5
A=25.1% C=24.7% G=24.8% T=25.4% N=0.000000% Other=0.000000%
Output least frequent oligs, direction=direct, seq_shift=0, seq_step=1
deviation multiplier=3.000000
#olig,total       olig       counter,expected       number,deviation,deviation
multiplier,unique sequences counter,norm deviate
Length 2  oligs=46673
TA      2174    2976.6     52.8    -15.2     32  0.046547
CG      2461    2858.0     51.8     -7.7     32  0.052692
GT      2609    2939.8     52.5     -6.3     32  0.055861
AC      2579    2893.8     52.1     -6.0     32  0.055219
GG      2662    2868.7     51.9     -4.0     32  0.056996

Length 3  oligs=46641
TAG       412     737.4     26.9    -12.1     32  0.008821
CTA       446     734.7     26.9    -10.7     32  0.009549
GTA       511     737.4     26.9     -8.4     32  0.010941
TAC       509     734.7     26.9     -8.4     31  0.010898
CGT       519     725.6     26.7     -7.7     32  0.011112
GGG       508     710.7     26.5     -7.7     32  0.010877
GTC       539     725.6     26.7     -7.0     32  0.011541
ACG       549     716.9     26.6     -6.3     32  0.011755
GAC       551     716.9     26.6     -6.2     32  0.011797
CCC       545     702.8     26.3     -6.0     32  0.011669
CGG       550     708.1     26.4     -6.0     32  0.011776
TTA       608     755.7     27.3     -5.4     32  0.013018
ATA       607     746.7     27.1     -5.2     31  0.012996
TAT       626     755.7     27.3     -4.8     32  0.013403
ACC       595     714.3     26.5     -4.5     32  0.012740
TAA       627     746.7     27.1     -4.4     32  0.013425
GGT       619     728.3     26.8     -4.1     32  0.013253
TCA       631     734.7     26.9     -3.9     32  0.013510
AGT       640     737.4     26.9     -3.6     32  0.013703
CCG       611     705.4     26.4     -3.6     32  0.013082
ACT       651     734.7     26.9     -3.1     32  0.013939

Length 4  oligs=46609
CTAG        73     182.0     13.5     -8.1     26  0.001563
GGGG        71     176.1     13.2     -7.9     24  0.001520
TAGG        83     182.7     13.5     -7.4     24  0.001777
CCTA        85     181.3     13.4     -7.2     26  0.001820
CGTA        92     182.0     13.5     -6.7     26  0.001970
TAGT       104     187.2     13.7     -6.1     26  0.002227
TTAG       105     187.2     13.7     -6.0     25  0.002248
```

```
ACGT        101     182.0     13.5     -6.0     29  0.002163
TACG        104     182.0     13.5     -5.8     22  0.002227
TAGA        108     185.0     13.6     -5.7     27  0.002312
TCTA        111     186.5     13.6     -5.5     27  0.002377
GGTA        110     182.7     13.5     -5.4     24  0.002355
ACTA        112     184.3     13.5     -5.3     29  0.002398
ACCC        106     176.3     13.3     -5.3     26  0.002270
GTCA        111     182.0     13.5     -5.3     26  0.002377
TAAC        113     184.3     13.5     -5.3     29  0.002419
CTAT        115     186.5     13.6     -5.2     29  0.002462
ATAG        115     185.0     13.6     -5.2     26  0.002462
CGGT        111     179.8     13.4     -5.1     30  0.002377
CGTC        111     179.1     13.4     -5.1     29  0.002377
CGGG        109     175.4     13.2     -5.0     29  0.002334
GATA        118     185.0     13.6     -4.9     27  0.002526
TATC        120     186.5     13.6     -4.9     30  0.002569
TACC        116     181.3     13.4     -4.9     26  0.002484
TAGC        117     182.0     13.5     -4.8     27  0.002505
TTAC        121     186.5     13.6     -4.8     28  0.002591
GTAG        119     182.7     13.5     -4.7     28  0.002548
ATAC        123     184.3     13.5     -4.5     26  0.002634
GGGT        121     180.4     13.4     -4.4     26  0.002591
CCCT        120     178.4     13.3     -4.4     29  0.002569
CGCG        117     174.8     13.2     -4.4     26  0.002505
GGTC        122     179.8     13.4     -4.3     29  0.002612
CTAA        126     184.3     13.5     -4.3     31  0.002698
GACC        120     177.0     13.3     -4.3     27  0.002569
TAAG        127     185.0     13.6     -4.3     30  0.002719
GTCT        127     184.2     13.5     -4.2     30  0.002719
CTTA        129     186.5     13.6     -4.2     31  0.002762
GTAA        128     185.0     13.6     -4.2     28  0.002741
ACGG        122     177.6     13.3     -4.2     30  0.002612
GACT        126     182.0     13.5     -4.2     31  0.002698
TCAT        130     186.5     13.6     -4.1     29  0.002783
AGAC        125     179.8     13.4     -4.1     28  0.002676
GTAT        132     187.2     13.7     -4.0     25  0.002826
CCCG        121     174.1     13.2     -4.0     28  0.002591
TACT        132     186.5     13.6     -4.0     29  0.002826
TGAC        129     182.0     13.5     -3.9     30  0.002762
CCGG        123     174.8     13.2     -3.9     27  0.002634
ACCG        125     177.0     13.3     -3.9     29  0.002676
ATTA        136     189.6     13.7     -3.9     29  0.002912
CCCC        123     173.5     13.1     -3.8     25  0.002634
AGTC        132     182.0     13.5     -3.7     26  0.002826
GTAC        132     182.0     13.5     -3.7     26  0.002826
CTAC        132     181.3     13.4     -3.7     31  0.002826
TCAC        132     181.3     13.4     -3.7     30  0.002826
CATA        135     184.3     13.5     -3.6     27  0.002890
AGTA        137     185.0     13.6     -3.5     29  0.002933
GCGT        136     179.8     13.4     -3.3     29  0.002912
GCTA        138     182.0     13.5     -3.3     28  0.002955
TCGT        140     184.2     13.5     -3.3     31  0.002998
GTTA        143     187.2     13.7     -3.2     29  0.003062
GAGT        140     182.7     13.5     -3.2     29  0.002998
TCGG        138     179.8     13.4     -3.1     31  0.002955
```

Detailed description for output data:

The program version and name are shown in the first string:

```
Oligs 1.6  Copyright (c) 2005-2006 Softberry
```

```
Num seqs=32 Nucleotides=46705 Average seq length=1459.5
```

```
A=25.1% C=24.7% G=24.8% T=25.4% N=0.000000% Other=0.000000%
```

Further there is an information on input file:
Number of fasta-sequences – 32
Number of nucleotides – 46705
Average length of sequence - 1459.5
Percentage of 'A' - 25.1
Percentage of 'C' - 24.7
Percentage of 'G' - 24.8
Percentage of 'T' - 25.4
Percentage of 'N' - 0.0
Percentage of other letters (except A,C,G,T,N) - 0.0

```
Output least frequent oligs, direction=direct, seq_shift=0, seq_step=1
deviation multiplier=3.000000
```

Further there are defined input parameters:
To show the most rare oligos - Output least frequent oligs.
Process the direct chain only - direction=direct
The "Shift" parameter – 0
The "Step" parameter – 1
Defined deviation multiplier range - 3.0
#olig,total olig counter,expected number,deviation,deviation multiplier,unique sequences counter,norm deviate
Further there is a hint for table of oligos on each column:
1 column - the specific oligo (olig)
2 column - the counter of this oligo, i.e. how much times this oligo occurs (total olig counter)
3 column - the expected counter mean value, i.e. expected average number of oligos (expected number)
4 column - the deviation of the current oligo (deviation)
5 column - the value of deviation multiplier for the current oligo (deviation multiplier) Note that in this example the value for deviation multipler range was set to 3.0. And since the mode to output the rarest oligos was chosen, the values in 5 column will be less or equal to -3.0.
6 column - the number of sequences containing the current oligo (unique sequences counter).
7 column - normalized deviation of the current oligo (norm deviate).
For more details on how various values are calculated see chapter "algorithm".

```
Length 3  oligs=46641
```

Further there are tables of oligos of different length.
Example for table of oligos of length 3
Here the length of the current oligo (Length 3) and total number of oligos of this length (oligs=46641) are shown.

```
TAG        412     737.4      26.9    -12.1      32  0.008821
CTA        446     734.7      26.9    -10.7      32  0.009549
GTA        511     737.4      26.9     -8.4      32  0.010941
```

Further there is the table with 5 column's values sorted by descending.
If it will be chosen the parameter to output the most frequent oligos, the values in 5 column will be sorted by ascending.
Description of values is shown earlier in the text.
The first string description.
1 column - The current oligo 'TAG'
2 column - The counter of the current oligo is 412
3 column - The expected oligo's mean is 737.4
4 column - The deviation for the current oligo is 26.9
5 column - The value for deviation multiplier for the current oligo is -12.1
6 column - The total number of sequences containing the current oligo is 32
7 column - Normalized deviation is 0.008821

Parameters:

Input	
Sequences set	Place your Input file in FASTA format.
Packed file	Input file is packed file (nucfile, nf).
Output	
Result	Name of the output file.
Print all oligs	Print all oligs, ignore conditions.
Options	
Frequency	Most frequent or least frequent: **most frequent (default)** **least frequent**
Minimal olig length	Minimal olig length.
Maximal oligs length	Maximal oligs length.
Scan chain	Scan target sequence in different chain: **In direct chain only (default)** **In reverse chain only** **In both chains**
Deviation multiplier fence	Use the value 3.0 to output 5% of oligos.
Shift in sequence	Shift in sequence, default value is 0.
Step in sequence	Step in sequence (default value is 1).

Oligs2

Search for such oligos (4-nucleotide oligos), that occur often in the 1st file and differ significantly in number on comparison of the 1st and 2nd files with sequences.

Input data

The input file should be in FASTA format and may contain several sequences. Alphabet. The allowed symbols: "ACGTUacgtu" and "NnyYrRBbDdHhKkWwSsMmVv". The symbols to be skipped: "0123456789; \n\r\t\0-". All other symbols are not allowed.

Input parameters

The program processes all oligonucleotides of length L. The L value runs all values in L1 to L2 range.

Minimal olig length (L1) - Minimal olig length

Minimal olig length (L2) - Minimal olig length

Restrictions for L1, L2: 1<=L1 && L1<=L2 && L2<=13.

Computer must have enough memory installed, and the memory size depends on oligo's length.

Input file 1 - The first input file in FASTA-format.

Input file 2 - The second input file in FASTA-format.

Coefficient k defines which one of these two files is most important at sorting the found oligos. It inflicts the sorting order for found oligos only. The default value 1.0 means the equal importance. If the k value is greater than 1.0, it means that the first file is more important, otherwise the second file is more important.

Coefficient k - Which one of the input files is more important for oligo (default 1.0)

Output file - Output file's name.

Algorithm

For the 1st input file the oligs program searches for the most frequent oligos at deviation multiplier = 0.0. The result is saved in temporary file.
For the 2nd input file the oligs program is run with "Print all oligs" option to find all oligos. The result is saved in temporary file.
It is important to search for definitely all oligos since an oligo existing in the 1st file may be represented in small amounts in the 2nd file also, and thus it could be problematic to compare the number of oligos in different files correctly.
For every oligo in the 1st temporary file the program searches for counterpart in the 2nd temporary file. For each oligo (taken from the 1st file) the program calculates the "sorter" value.
The ratio of nucleotides number between files - div_sum_len:
div_sum_len= number of nucleotides in the 1st file/number of nucleotides in the 2nd file;
Coefficient k - input parameter.
olig1_count - how many times oligo occurs in the 1st file.
olig2_count - how many times oligo occurs in the 2nd file.
z= 0.5*olig1_count*(1+k*olig1_count/(olig2_count*div_sum_len))
The "derivation multiplier" value for oligo from the 1st temporary file - olig1_derivat_mult.
sorter=olig1_derivat_mult*z;
The program prints the title from 1st temporary file, then the title from 2nd one, and then all oligos in "sorter" descend order.

Output data

Example for program output:

```
Oligs2 1.1  Copyright (c) 2005-2006 Softberry
Num seqs=11 Nucleotides=12191 Average seq length=1108.3
A=25.4% C=23.9% G=25.0% T=25.1% N=0.623411% Other=0.000000%
Output most frequent oligs, direction=direct, seq_shift=0, seq_step=1
deviation multiplier=0.000000
Num seqs=17 Nucleotides=13702 Average seq length=806.0
A=28.8% C=21.4% G=21.8% T=28.0% N=0.000000% Other=0.000000%
Output most frequent oligs, direction=direct, seq_shift=0, seq_step=1
all by distant
#olig,total  olig  counter1,expected  number1,unique  sequences  counter1,total
olig counter2,
unique sequences counter2,norm deviate1,norm deviate 2,sorter
Length 2
TG       899     764.6     11        954     17  0.073743  0.069625    4627.9
CA       873     738.4     11        927     17  0.071610  0.067654    4582.5
GC       832     727.2     11        830     17  0.068247  0.060575    3538.7
TT       871     768.9     11       1296     17  0.071446  0.094585    2905.0
AA       875     784.0     11       1414     17  0.071774  0.103197    2522.1
GA       842     772.1     11        759     17  0.069067  0.055393    2459.4
TC       788     731.2     11        744     17  0.064638  0.054299    1898.7
AT       804     776.4     11       1067     17  0.065950  0.077872     742.5
AG       786     772.1     11        755     17  0.064474  0.055101     426.4

Length 3
CTG       260     182.5     11       210     17  0.021327  0.015326    1803.2
TTT       278     193.0     11       482     17  0.022804  0.035177    1420.5
CAG       247     184.3     11       207     17  0.020261  0.015107    1358.9
CCA       237     176.3     11       232     17  0.019441  0.016932    1171.0
TGC       242     182.5     11       261     17  0.019851  0.019048    1087.2
```

```
TGG       246     190.9      11        242      17  0.020179  0.017662    1054.1
AAA       268     198.7      11        568      17  0.021983  0.041454    1025.3
GGA       239     192.7      11        183      17  0.019605  0.013356    1002.7
TCC       222     174.6      11        167      17  0.018210  0.012188     996.6
TTC       235     183.6      11        236      17  0.019277  0.017224     946.2
GCA       234     184.3      11        236      17  0.019194  0.017224     915.3
GAA       243     195.7      11        239      17  0.019933  0.017443     885.2
AGC       229     184.3      11        207      17  0.018784  0.015107     847.7
GCT       227     182.5      11        222      17  0.018620  0.016202     805.0
ATC       223     185.4      11        204      17  0.018292  0.014888     695.8
CAT       224     185.4      11        233      17  0.018374  0.017005     675.8
GAG       223     192.7      11        161      17  0.018292  0.011750     627.2
CAA       228     187.2      11        315      17  0.018702  0.022989     620.2
ATG       226     193.8      11        247      17  0.018538  0.018027     527.2
AAG       227     195.7      11        273      17  0.018620  0.019924     505.0
GCC       202     173.6      11        215      17  0.016570  0.015691     456.8
TCA       210     185.4      11        210      17  0.017226  0.015326     401.4
GAT       214     193.8      11        204      17  0.017554  0.014888     349.7
CGA       202     184.3      11        184      17  0.016570  0.013429     293.3
ATT       216     194.9      11        341      17  0.017718  0.024887     277.3
CTT       202     183.6      11        245      17  0.016570  0.017881     272.4
GTG       207     190.9      11        205      17  0.016980  0.014961     265.2
TGA       207     193.8      11        206      17  0.016980  0.015034     220.4
TTG       206     191.9      11        292      17  0.016898  0.021311     184.7
TGT       204     191.9      11        245      17  0.016734  0.017881     177.7
AGG       198     192.7      11        161      17  0.016241  0.011750      94.3
CGC       177     173.6      11        160      17  0.014519  0.011677      59.6
ACA       190     187.2      11        248      17  0.015585  0.018100      35.4
AAT       200     196.8      11        340      17  0.016406  0.024814      33.2
GGC       183     181.5      11        202      17  0.015011  0.014742      18.5
```

Detailed description for output data:

The program version and name are shown in the first string:

```
Oligs2 1.1  Copyright (c) 2005-2006 Softberry
Num seqs=11 Nucleotides=12191 Average seq length=1108.3
A=25.4% C=23.9% G=25.0% T=25.1% N=0.623411% Other=0.000000%
Output most frequent oligs, direction=direct, seq_shift=0, seq_step=1
deviation multiplier=0.000000
```

It is the title for first program run. It is information on 1st input file:
Number of fasta-sequences - 11
Number of nucleotides - 12191
Average length of sequence - 1108.3

```
Num seqs=17 Nucleotides=13702 Average seq length=806.0
A=28.8% C=21.4% G=21.8% T=28.0% N=0.000000% Other=0.000000%
Output most frequent oligs, direction=direct, seq_shift=0, seq_step=1
all by distant
```

It is the title for second program run. It is information on 2nd input file:
Number of fasta-sequences - 17
Number of nucleotides - 13702
Average length of sequence - 806.0

```
#olig,total  olig  counter1,expected  number1,unique  sequences  counter1,total
olig counter2,
unique sequences counter2,norm deviate1,norm deviate 2,sorter
```

Further the hint for table of oligos by columns is sown:
1 column - certain oligo (olig)

2 column - counter for current oligo in the 1st file, i.e. how many times this oligo occurs in the 1st file (total olig counter1)
3 column - expected counter mean for the 1st file, i.e. an expected average number of oligos in the 1st file (expected number1)
4 column - number of sequences form the 1st file, in which this oligo occurs (unique sequences counter1).
5 column - counter for current oligo in the 2nd file, i.e. how many times this oligo occurs in the 2nd file (total olig counter2)
6 column - number of sequences form the 2nd file, in which this oligo occurs (unique sequences counter2)
7 column - normalized deviation of this oligo for the 1st file (norm deviate1).
8 column - normalized deviation of this oligo for the 2nd file (norm deviate2).
9 column - "sorter" value for current oligo (sorter).
For more details on how various values are calculated see chapter "algorithm".
Length 3

Further there are tables of oligos of different length.
Example for table of oligos of length 3
Here the length of the current oligo (Length 3)

```
CTG        260     182.5      11        210      17  0.021327  0.015326    1803.2
TTT        278     193.0      11        482      17  0.022804  0.035177    1420.5
CAG        247     184.3      11        207      17  0.020261  0.015107    1358.9
```

Further there is the table sorted by descend of 9th column.
Columns description is above in the text.
Description of the first string:
1 column - certain oligo 'CTG'
2 column - counter for current oligo in the 1st file 260
3 column - expected counter mean for the 1st file 182.5
4 column - number of sequences form the 1st file, in which this oligo occurs, 11
5 column - counter for current oligo in the 2nd file 210
6 column - number of sequences form the 2nd file, in which this oligo occurs 17
7 column - normalized deviation of this oligo for the 1st file 0.021327
8 column - normalized deviation of this oligo for the 2nd file 0.015326
9 column - "sorter" value for current oligo 1803.2

Parameters:

Input	
Sequences set 1	The first input file in FASTA-format.
Sequences set 2	The second input file in FASTA-format.
Output	
Result file	Output file's name.
Options	
Minimal olig length	Minimal olig length.
Maximal olig length (L2)	Maximal olig length.
Coefficient k	Which one of the input files is more important for oligo (default 1.0)

OligsR

The program makes the statistical calculations on redundant oligos (15-mer oligos) and displays the oligos, that differ from expected mean significantly.

Input data

The input file should be in FASTA format and may contain several sequences. Alphabet. The allowed symbols: "ACGTUacgtu" and "NnyYrRBbDdHhKkWwSsMmVv". The symbols to be skipped: "0123456789; \n\r\t\0-". All other symbols are not allowed.

Input parameters

The program processes all oligonucleotides of length L. The L value runs all values in L1 to L2 range.

Minimal olig length (L1) - Minimal olig length

Minimal olig length (L2) - Minimal olig length

Restrictions for L1, L2: 1<=L1 && L1<=L2 && L2<=6.

Computer must have enough memory installed, and the memory size depends on oligo's length.

Input file - Input file in FASTA-format.

The special mode to print all oligos ignoring any additional conditions. While in this mode the very big output file can be generated.

Print all oligs - Print all oligs, ignore conditions

The program can process not only the given sequence but simultaneously build and process the reverse sequence.

Scan target sequence in different chain - Scan target sequence in different chain:
In direct chain only (default)
In reverse chain only
In both chains

Similarly to normal distribution, the program can output either most frequent oligos or most rare ones. The following parameter is used for this:

Frequency - Most frequent or least frequent:
most frequent (default)
least frequent

To determine which oligos must be output and which ones must not, the value for deviation multiplier range should be defined.

Deviation multiplier is difference between number of oligos and expected number of oligos in sigma units. For more details see the algorithm description chapter.

Deviation multiplier fence - Use the value 3.0 to output 5% of oligos.

On oligo output, an additional filtering is made. For each oligo, the percentage of letters 'N' in relation to all letters of oligo is calculated. Oligos, for which this percentage does not exceed the "Percent of N" parameter, are output.

Percent of N - Olig have no more # % of 'N', default is 100.

Output file - Output file name.

The "shift" parameter sets the value (in nucleotides) of shifting from the sequence start to the position from which oligos are to be generated. If there are several sequences in a file, the shift value affects each of them. The default value is 0.

Shift in sequence - Shift in sequence, default value is 0.

The "step" parameter sets the value (in nucleotides) of shifting for generating oligos. In order to get all oligos, this parameter should be set to 1, which is default value.

Step in sequence - Step in sequence (default value is 1)

Sometime it's necessary to check all three reading frames. To do this run the program three times with the following values for "shift" and "step":

1) step=3 shift=0
2) step=3 shift=1
3) step=3 shift=2

Input sequences may be either in FASTA format or in specially packed format. The "Softberry" products frequently used to pack large chromosomes into its own "nucfile" or nf format. Sequence file, in this case, has the .nf extension.

If the "Packed file" parameter is not defined the program consider the input file as one in FASTA format. Otherwise the input file format is considered as "nucfile".

Packed file - Input file is packed file (nucfile, nf).

The FASTA file can be converted to the nucfile one using the cvtseq utility. For example, to convert the FASTA file chr22.fa to the nucfile chr22.nf, use the following command string:

```
cvtseq chr22.fa chr22.nf -fi -do -t "chr22" -n5gc
```

Use the following command to check the information on a packed file:

```
cvtseq chr22.nf -e
```

Command output:

```
filename: chr22.nf
pack_mode: PACK_5

size: 49476972 from: 0 nonstandard: 1
title_size: 5 title: chr22
```

Algorithm

For each defined L the array that contains the number of oligos is built. The sequential number of oligo is used as an index for this array. The total number of oligos is a value of the array.

Further, using this array and defined parameters, program builds the table of oligos that contains more information (mean, deviation multiplier etc). This table is printed into output file.

Total number of all oligos - oligs_sum_count.
Total number of nucleotides - seqs_sum_length.
The oligo's frequency is a multiplication of frequencies of nucleotides it consists of.
The expected mean of the counter (that is equal to oligo's mean) is calculated by the following way:
average= oligs_sum_count*frequence;
Deviation is calculated with use of formula:
deviation = sqrt(oligs_sum_count*frequence*(1-frequence));
The oligo's counter - olig_count - describes how much times this oligo occurs in a sequence.
Deviation multiplier is calculated with use of formula:
Deviation_multiplier= (olig_count-average)/deviation;
Normalized deviation (norm deviate) of the given oligo is calculated with use of formula:
Norm_deviate= olig_count/seqs_sum_length;

Output data

Example for program output:

```
Oligsr 1.4  Copyright (c) 2005-2006 Softberry
Num seqs=32 Nucleotides=46705 Average seq length=1459.5
A=25.1% C=24.7% G=24.8% T=25.4%
AC=49.8% AG=49.9% AT=50.5% CG=49.5% CT=50.1% GT=50.2%
ACG=74.6% ACT=75.2% AGT=75.3% CGT=74.9% N=100.0%
Output most frequent oligs, direction=direct, deviation multiplier=10.000000,
no more 50.0 % of 'N'
#olig,total       olig       counter,expected       number,deviation,deviation
multiplier,unique sequences counter,norm deviate
Length 1

Length 2
TK      6906    5952.4     72.1     13.2     32  0.147864
TG      3544    2939.8     52.5     11.5     32  0.075881
MA      6654    5834.8     71.5     11.5     32  0.142469
GC      3409    2858.0     51.8     10.6     32  0.072990

Length 3
TKB      5574    4455.2     63.5     17.6     32  0.119345
VMA      5390    4349.4     62.8     16.6     32  0.115405
TKS      3731    2943.9     52.5     15.0     32  0.079884
YTK      3772    2980.5     52.8     15.0     32  0.080762
TGS      1993    1453.9     37.5     14.4     32  0.042672
TBB      7724    6647.3     75.5     14.3     32  0.165378
VMW      9944    8751.5     84.3     14.1     32  0.212911
MMA      3639    2903.8     52.2     14.1     32  0.077915
MAR      3639    2909.2     52.2     14.0     32  0.077915
VVA      7555    6514.6     74.9     13.9     32  0.161760
WKB     10034    8857.0     84.7     13.9     32  0.214838
TKY      3711    2980.5     52.8     13.8     32  0.079456
BTK      5330    4455.2     63.5     13.8     32  0.114121
YTB      5315    4447.0     63.4     13.7     32  0.113799
HTK      5343    4473.6     63.6     13.7     32  0.114399
VAR      5214    4357.4     62.9     13.6     32  0.111637
TKK      3706    2986.0     52.9     13.6     32  0.079349
TGB      2820    2200.3     45.8     13.5     32  0.060379
GCH      2754    2148.0     45.3     13.4     32  0.058966
WGC      1942    1442.6     37.4     13.4     32  0.041580
TKN      6904    5948.3     72.0     13.3     32  0.147821
NTK      6901    5948.3     72.0     13.2     32  0.147757
CWG      1936    1442.6     37.4     13.2     32  0.041452
GCW      1936    1442.6     37.4     13.2     32  0.041452
YKB      9894    8786.4     84.4     13.1     32  0.211840
RMA      3590    2909.2     52.2     13.0     32  0.076865
MAV      5157    4349.4     62.8     12.9     32  0.110416
RMW      6771    5853.7     71.5     12.8     32  0.144974
SMA      3551    2885.7     52.0     12.8     32  0.076030
WKS      6767    5852.6     71.5     12.8     32  0.144888
SCW      3540    2879.8     52.0     12.7     32  0.075795
YKS      6708    5806.0     71.3     12.7     32  0.143625
SWG      3548    2890.5     52.1     12.6     32  0.075966
MAA      1937    1463.7     37.7     12.6     32  0.041473
WGS      3545    2890.5     52.1     12.6     32  0.075902
VMR      9694    8645.0     83.9     12.5     32  0.207558
TBS      5180    4392.5     63.1     12.5     32  0.110909
DGC      2716    2150.6     45.3     12.5     32  0.058152
TGC      1057     725.6     26.7     12.4     32  0.022631
VMD     14248   13047.1     96.9     12.4     32  0.305064
HKS      9744    8714.7     84.2     12.2     32  0.208629
```

```
SCA      1886    1431.2      37.2      12.2      32  0.040381
YTG      1932    1472.0      37.8      12.2      32  0.041366
BTG      2755    2200.3      45.8      12.1      32  0.058987
TBY      5213    4447.0      63.4      12.1      32  0.111615
HTB      7583    6674.9      75.6      12.0      32  0.162359
HKB     14354   13188.3      97.3      12.0      32  0.307333
VWG      5106    4356.5      62.8      11.9      32  0.109324
SMW      6654    5806.4      71.3      11.9      32  0.142469
AAA      1058     737.7      26.9      11.9      32  0.022653
VAD      7463    6576.3      75.2      11.8      32  0.159790
MAD      5129    4390.6      63.1      11.7      32  0.109817
SMD      9638    8656.5      84.0      11.7      32  0.206359
VAA      2723    2192.3      45.7      11.6      32  0.058302
TGN      3542    2937.8      52.5      11.5      32  0.075838
NTG      3542    2937.8      52.5      11.5      32  0.075838
TGV      2715    2191.4      45.7      11.5      32  0.058131
NMA      6648    5830.8      71.4      11.4      32  0.142340
MAN      6647    5830.8      71.4      11.4      32  0.142319
KSC      3450    2862.0      51.8      11.3      32  0.073868
TTK      1943    1511.3      38.2      11.3      32  0.041602
CWS      3466    2879.8      52.0      11.3      32  0.074210
SMR      6535    5735.8      70.9      11.3      32  0.139921
VCA      2667    2157.1      45.4      11.2      32  0.057103
MWG      3494    2908.6      52.2      11.2      32  0.074810
HTG      2719    2209.4      45.9      11.1      32  0.058216
RVA      5055    4357.4      62.9      11.1      32  0.108233
MVA      5045    4349.4      62.8      11.1      32  0.108018
KSH      9645    8714.7      84.2      11.1      32  0.206509
WKY      6717    5925.3      71.9      11.0      32  0.143818
SVA      5010    4322.3      62.6      11.0      32  0.107269
GMW      3481    2908.6      52.2      11.0      32  0.074532
TSC      1858    1448.5      37.5      10.9      32  0.039782
TGY      1884    1472.0      37.8      10.9      32  0.040338
TTB      2754    2254.9      46.3      10.8      32  0.058966
HGC      2632    2148.0      45.3      10.7      32  0.056354
KSY      6568    5806.0      71.3      10.7      32  0.140627
KGC      1831    1433.7      37.3      10.7      32  0.039204
GCN      3407    2856.1      51.8      10.6      32  0.072947
KSM      6527    5770.7      71.1      10.6      32  0.139749
NGC      3406    2856.1      51.8      10.6      32  0.072926
KBB     14164   13133.9      97.1      10.6      32  0.303265
TKC      1868    1469.2      37.7      10.6      32  0.039996
MAM      3455    2903.8      52.2      10.6      32  0.073975
CTG      1005     725.6      26.7      10.5      32  0.021518
KBY      9669    8786.4      84.4      10.5      32  0.207023
TBC      2669    2192.1      45.7      10.4      32  0.057146
VVM     13931   12924.7      96.7      10.4      32  0.298276
VWK      9698    8821.1      84.6      10.4      32  0.207644
TSS      3442    2902.5      52.2      10.3      32  0.073697
TKG      1863    1474.7      37.8      10.3      32  0.039889
VAV      7283    6514.6      74.9      10.3      32  0.155936
MMR      6501    5771.8      71.1      10.3      32  0.139193
YTS      3475    2938.5      52.5      10.2      32  0.074403
DSC      4930    4293.3      62.4      10.2      32  0.105556
BTB      7412    6647.3      75.5      10.1      32  0.158698
WGB      5012    4374.3      63.0      10.1      32  0.107312
CWK      3450    2920.9      52.3      10.1      32  0.073868
WKC      3450    2920.9      52.3      10.1      32  0.073868
VCW      4972    4340.4      62.7      10.1      32  0.106455
RAA      1844    1466.4      37.7      10.0      32  0.039482
VHD     20770   19703.0     106.7      10.0      32  0.444706
```

Detailed description for output data:

The program version and name are shown in the first string:

```
Oligsr 1.4  Copyright (c) 2005-2006 Softberry
Num seqs=32 Nucleotides=46705 Average seq length=1459.5
A=25.1% C=24.7% G=24.8% T=25.4%
AC=49.8% AG=49.9% AT=50.5% CG=49.5% CT=50.1% GT=50.2%
ACG=74.6% ACT=75.2% AGT=75.3% CGT=74.9% N=100.0%
```

Further there is an information on input file:

Number of fasta-sequences - 32
Number of nucleotides - 46705
Average length of sequence - 1459.
Percentage of letters 'A' - 25.1
Percentage of letters 'C' - 24.7
Percentage of letters 'G' - 24.8
Percentage of letters 'T' - 25.4
Percentage of letters 'A or C' - 49.8
Percentage of letters 'A or G' - 49.9
Percentage of letters 'A or T' - 50.5
Percentage of letters 'C or G' - 49.5
Percentage of letters 'C or T' - 50.1
Percentage of letters 'G or T' - 50.2
Percentage of letters 'A or ╤ or G' - 74.6
Percentage of letters 'A or ╤ or T' - 75.2
Percentage of letters 'A or G or T' - 75.3
Percentage of letters 'C or G or T' - 74.9
Percentage of letters 'A or C or G or T' - 100.0

```
Output most frequent oligs, direction=direct, deviation multiplier=10.000000,
no more 50.0 % of 'N'
```

Further there are defined input parameters:
To output the most frequent oligos - Output most frequent oligs.
To process the direct chain only - direction=direct
Defined range for deviation multiplier - 10.0
To output oligos containing not more than 50% of letters 'N'.

```
#olig, total olig counter, expected number, deviation, deviation multiplier,
unique sequences counter, norm deviate
```

Further there is the hint on table of oligos by columns:

1 column -certain oligo (olig)
2 column - counter for current oligo, i.e. how many times this oligo occurs (total olig counter)
3 column - expected counter mean, i.e. an expected average number of oligos (expected number)
4 column - deviation of current oligo (deviation)
5 column -deviation multiplier value for current oligo (deviation multiplier)
To remind, in given example the range for deviation multiplier was set to 3.0. And since the option to output the most rare oligos was selected, the values in 5th column will be less or equal to -3.0.
6 column - number of sequences, in which this oligo occurs.
7 column - normalized deviation of this oligo.
For more details on values calculation see the chapter "Algorithm"

```
Length 3
```

Further there are tables of oligos with various length values.

Hereafter is an example of the table with oligos of length 3.
The length of examined oligo (Length 3) is shown.

```
TKB      5574    4455.2     63.5      17.6      32  0.119345
VMA      5390    4349.4     62.8      16.6      32  0.115405
TKS      3731    2943.9     52.5      15.0      32  0.079884
```

Further there is a table sorted by 5th column descend.

If the option to output the most frequent oligos is on, the table will be sorted by 5th column ascend.
Description of values in columns is above in the text.

The first string description:

1 column - certain oligo 'TKB'
2 column - counter for current oligo 5574
3 column - expected mean for oligo 4455.2
4 column - deviation of current oligo 63.5
5 column - deviation multiplier value for current oligo -17.6
6 column - number of sequences, in which this oligo occurs 32
7 column - normalized deviation of this oligo 0.119345

Parameters:

Input	
Sequences set	Place your Input file in FASTA format.
Packed file	Input file is packed file (nucfile, nf).
Output	
Result	Name of the output file.
Print all oligs	Print all oligs, ignore conditions.
Print oligs by deviation	Use the value 3.0 to output 5% of oligos.
Options	
Frequency	Most frequent or least frequent: **most frequent (default)** **least frequent**
Minimal olig length	Minimal olig length.
Maximal oligs length	Maximal oligs length.
Percents of N	Olig have no more # % of 'N', default is 100.
Scan chain	Scan target sequence in different chain: **In direct chain only (default)** **In reverse chain only** **In both chains**
Shift in sequence	Shift in sequence, default value is 0.
Step in sequence	Step in sequence (default value is 1).

Primer3

Primer3 picks primers for PCR reactions, considering as criteria:

- oligonucleotide melting temperature, size, GC content, and primer-dimer possibilities,
- PCR product size,
- positional constraints within the source sequence, and
- miscellaneous other constraints.

All of these criteria are user-specifiable as constraints, and some are specifiable as terms in an objective function that characterizes an optimal primer pair.
This product includes software developed by the Whitehead Institute for Biomedical Research.

Copyright Notice and Disclaimer:

Use of this software should be cited in publications as

Rozen, S., Skaletsky, H. "Primer3 on the WWW for general users and for biologist programmers." In S. Krawetz and S. Misener, eds. Bioinformatics Methods and Protocols in the series Methods in Molecular Biology. Humana Press, Totowa, NJ, 2000, pages 365-386.

Code available at http://fokker.wi.mit.edu/primer3/

Primer3's design is heavily based on an earlier implementation of a similar program: Primer 0.5 (Steve Lincoln, Mark Daly, and Eric S. Lander). Lincoln Stein championed the idea of making the Primer3 engine a software component.

Primer3 Input Help

Cautions

Some of the most important issues in primer picking can be addressed only before using Primer3. These are sequence quality (including making sure the sequence is not vector and not chimeric) and avoiding repetitive elements.
Techniques for avoiding problems include a thorough understanding of possible vector contaminants and cloning artifacts coupled with database searches using blast, fasta, or other similarity searching program to screen for vector contaminants and possible repeats. Repbase (J. Jurka, A.F.A. Smit, C. Pethiyagoda, and others, 1995-1996) ftp://ftp.ncbi.nih.gov/repository/repbase) is an excellent source of repeat sequences and pointers to the literature. Primer3 now allows you to screen candidate oligos against a Mispriming Library (or a Mishyb Library in the case of internal oligos).

Sequence quality can be controlled by manual trace viewing and quality clipping or automatic quality clipping programs. Low- quality bases should be changed to N's or can be made part of Excluded Regions. The beginning of a sequencing read is often problematic because of primer peaks, and the end of the read often contains many low-quality or even meaningless called bases. Therefore when picking primers from single-pass sequence it is often best to use the Included Region parameter to ensure that Primer3 chooses primers in the high quality region of the read. In addition, Primer3 takes as input a Sequence Quality list for use with those base calling programs such as Phred that output this information.

Source Sequence

The sequence from which to select primers or hybridization oligos.

Sequence Id

An identifier that is reproduced in the output to enable you to identify the chosen primers.

Targets

If one or more Targets is specified then a legal primer pair must flank at least one of them. A Target might be a simple sequence repeat site (for example a CA repeat) or a single-base-pair polymorphism. The value should be a space-separated list of
start,length

pairs where `start` is the index of the first base of a Target, and `length` is its length.

Excluded Regions

Primer oligos may not overlap any region specified in this tag. The associated value must be a space-separated list of

```
start,length
```

pairs where `start` is the index of the first base of the excluded region, and `length` is its length. This tag is useful for tasks such as excluding regions of low sequence quality or for excluding regions containing repetitive elements such as ALUs or LINEs.

Product Size Range

A list of product size ranges, for example

```
150-250 100-300 301-400
```

Primer3 first tries to pick primers in the first range. If that is not possible, it goes to the next range and tries again. It continues in this way until it has either picked all necessary primers or until there are no more ranges. For technical reasons this option makes much lighter computational demands than the Product Size option.

Product Size

Minimum, Optimum, and Maximum lengths (in bases) of the PCR product. Primer3 will not generate primers with products shorter than Min or longer than Max, and with default arguments Primer3 will attempt to pick primers producing products close to the Optimum length.

Number To Return

The maximum number of primer pairs to return. Primer pairs returned are sorted by their "quality", in other words by the value of the objective function (where a lower number indicates a better primer pair). Caution: setting this parameter to a large value will increase running time.

Max 3' Stability

The maximum stability for the five 3' bases of a left or right primer. Bigger numbers mean more stable 3' ends. The value is the maximum delta G for duplex disruption for the five 3' bases as calculated using the nearest neighbor parameters published in Breslauer, Frank, Bloeker and Marky, Proc. Natl. Acad. Sci. USA, vol 83, pp 3746-3750. Rychlik recommends a maximum value of 9 (Wojciech Rychlik, "Selection of Primers for Polymerase Chain Reaction" in BA White, Ed., "Methods in Molecular Biology, Vol. 15: PCR Protocols: Current Methods and Applications", 1993, pp 31-40, Humana Press, Totowa NJ).

Max Mispriming

The maximum allowed weighted similarity with any sequence in Mispriming Library. Default is 12.

Pair Max Mispriming

The maximum allowed sum of similarities of a primer pair (one similarity for each primer) with any single sequence in Mispriming Library. Default is 24. Library sequence weights are not used in computing the sum of similarities.

Primer Size

Minimum, Optimum, and Maximum lengths (in bases) of a primer oligo. Primer3 will not pick primers shorter than Min or longer than Max, and with default arguments will attempt to pick primers close with size close to Opt. Min cannot be smaller than 1. Max cannot be larger than 36. (This limit is governed by maximum oligo size for which melting-temperature calculations are valid.) Min cannot be greater than Max.

Primer T_m

Minimum, Optimum, and Maximum melting temperatures (Celsius) for a primer oligo. Primer3 will not pick oligos with temperatures smaller than Min or larger than Max, and with default conditions will try to pick primers with melting temperatures close to Opt.

Primer3 uses the oligo melting temperature formula given in Rychlik, Spencer and Rhoads, Nucleic Acids Research, vol 18, num 21, pp 6409-6412 and Breslauer, Frank, Bloeker and Marky, Proc. Natl. Acad. Sci. USA, vol 83, pp 3746-3750. Please refer to the former paper for background discussion.

Maximum T_m Difference

Maximum acceptable (unsigned) difference between the melting temperatures of the left and right primers.

Product T_m

The minimum, optimum, and maximum melting temperature of the amplicon. Primer3 will not pick a product with melting temperature less than min or greater than max. If Opt is supplied and the Penalty Weights for Product Size are non-0 Primer3 will attempt to pick an amplicon with melting temperature close to Opt.

The maximum allowed melting temperature of the amplicon. Primer3 calculates product T_m calculated using the formula from Bolton and McCarthy, PNAS 84:1390 (1962) as presented in Sambrook, Fritsch and Maniatis, Molecular Cloning, p 11.46 (1989, CSHL Press).

$T_m = 81.5 + 16.6(\log_{10}([Na+])) + .41*(\%GC) - 600/length$,

where [Na+] is the molar sodium concentration, (%GC) is the percent of Gs and Cs in the sequence, and length is the length of the sequence.

A similar formula is used by the prime primer selection program in GCG (http://www.gcg.com), which instead uses 675.0 / length in the last term (after F. Baldino, Jr, M.-F. Chesselet, and M.E. Lewis, Methods in Enzymology 168:766 (1989) eqn (1) on page 766 without the mismatch and formamide terms). The formulas here and in Baldino et al. assume Na+ rather than K+. According to J.G. Wetmur, Critical Reviews in BioChem. and Mol. Bio. 26:227 (1991) 50 mM K+ should be equivalent in these formulae to .2 M Na+. Primer3 uses the same salt concentration value for calculating both the primer melting temperature and the oligo melting temperature. If you are planning to use the PCR product for hybridization later this behavior will not give you the T_m under hybridization conditions.

Primer GC% Minimum, Optimum, and Maximum percentage of Gs and Cs in any primer.

Max Complementarity

The maximum allowable local alignment score when testing a single primer for (local) self-complementarity and the maximum allowable local alignment score when testing for complementarity between left and right primers. Local self-complementarity is taken to predict the tendency of primers to anneal to each other without necessarily causing self-priming in the PCR. The scoring system gives 1.00 for complementary bases, -0.25 for a match of any base (or N) with an N, -1.00 for a mismatch, and -2.00 for a gap. Only single-base-pair gaps are allowed. For example, the alignment

```
5' ATCGNA 3'
   || | |
3' TA-CGT 5'
```

is allowed (and yields a score of 1.75), but the alignment

```
5' ATCCGNA 3'
   ||  | |
3' TA--CGT 5'
```

is not considered. Scores are non-negative, and a score of 0.00 indicates that there is no reasonable local alignment between two oligos.

Max 3' Complementarity

The maximum allowable 3'-anchored global alignment score when testing a single primer for self-complementarity, and the maximum allowable 3'-anchored global alignment score when testing for complementarity between left and right primers. The 3'-anchored global alignment score is taken to predict the likelihood of PCR-priming primer-dimers, for example

```
5' ATGCCCTAGCTTCCGGATG 3'
             ||| |||||
          3' AAGTCCTACATTTAGCCTAGT 5'
```

or

```
5` AGGCTATGGGCCTCGCGA 3'
               ||||||
            3' AGCGCTCCGGGTATCGGA 5'
```

The scoring system is as for the Max Complementarity argument. In the examples above the scores are 7.00 and 6.00 respectively. Scores are non-negative, and a score of 0.00 indicates that there is no reasonable 3'-anchored global alignment between two oligos. In order to estimate 3'-anchored global alignments for candidate primers and primer pairs, Primer assumes that the sequence from which to choose primers is presented 5'->3'. It is nonsensical to provide a larger value for this parameter than for the Maximum (local) Complementarity parameter because the score of a local alignment will always be at least as great as the score of a global alignment.

Max Poly-X

The maximum allowable length of a mononucleotide repeat, for example AAAAAA.

Included Region

A sub-region of the given sequence in which to pick primers. For example, often the first dozen or so bases of a sequence are vector, and should be excluded from consideration. The value for this parameter has the form

start,length

where *start* is the index of the first base to consider, and *length* is the number of subsequent bases in the primer-picking region.

Start Codon Position

This parameter should be considered EXPERIMENTAL at this point. Please check the output carefully; some erroneous inputs might cause an error in Primer3. Index of the first base of a start codon. This parameter allows Primer3 to select primer pairs to create in-frame amplicons e.g. to create a template for a fusion protein. Primer3 will attempt to select an in-frame left primer, ideally starting at or to the left of the start codon, or to the right if necessary. Negative values of this parameter are legal if the actual start codon is to the left of available sequence. If this parameter is non-negative Primer3 signals an error if the codon at the position specified by this parameter is not an ATG. A value less than or equal to -10^6 indicates that Primer3 should ignore this parameter. Primer3 selects the position of the right primer by scanning right from the left primer for a stop codon. Ideally the right primer will end at or after the stop codon.

Mispriming Library

This selection indicates what mispriming library (if any) Primer3 should use to screen for interspersed repeats or for other sequence to avoid as a location for primers. The human and rodent libraries on the web page are adapted from Repbase (J. Jurka, A.F.A. Smit, C. Pethiyagoda, et al., 1995-1996) ftp://ftp.ncbi.nih.gov/repository/repbase). The human library is humrep.ref concatenated with simple.ref, translated to FASTA format. There are two rodent libraries. One is rodrep.ref translated to FASTA format, and the other is rodrep.ref concatenated with simple.ref, translated to FASTA format.

The *Drosophila* library is the concatenation of two libraries from the Berkeley Drosophila Genome Project:

1. A library of transposable elements The transposable elements of the Drosophila melanogaster euchromatin - a genomics perspective J.S. Kaminker, C.M. Bergman, B. Kronmiller, J. Carlson, R. Svirskas, S. Patel, E. Frise, D.A. Wheeler, S.E. Lewis, G.M.

Rubin, M. Ashburner and S.E. Celniker Genome Biology (2002) 3(12):research0084.1-0084.20, http://www.fruitfly.org/p_disrupt/datasets/ASHBURNER/D_mel_transposon_sequence_set.fasta

2. A library of repetitive DNA sequences http://www.fruitfly.org/sequence/sequence_db/na_re.dros.
Both were downloaded 6/23/04.

The contents of the libraries can be viewed at the following links:

- HUMAN (contains microsatellites)
- RODENT_AND_SIMPLE (contains microsatellites)
- RODENT (does not contain microsatellites)
- DROSOPHILA

CG Clamp

Require the specified number of consecutive Gs and Cs at the 3' end of both the left and right primer. (This parameter has no effect on the hybridization oligo if one is requested.)

Salt Concentration

The millimolar concentration of salt (usually KCl) in the PCR. Primer3 uses this argument to calculate oligo melting temperatures.

Annealing Oligo Concentration

The nanomolar concentration of annealing oligos in the PCR. Primer3 uses this argument to calculate oligo melting temperatures. The default (50nM) works well with the standard protocol used at the Whitehead/MIT Center for Genome Research--0.5 microliters of 20 micromolar concentration for each primer oligo in a 20 microliter reaction with 10 nanograms template, 0.025 units/microliter Taq polymerase in 0.1 mM each dNTP, 1.5mM MgCl2, 50mM KCl, 10mM Tris-HCL (pH 9.3) using 35 cycles with an annealing temperature of 56 degrees Celsius. This parameter corresponds to 'c' in Rychlik, Spencer and Rhoads' equation (ii) (Nucleic Acids Research, vol 18, num 21) where a suitable value (for a lower initial concentration of template) is "empirically determined". The value of this parameter is less than the actual concentration of oligos in the reaction because it is the concentration of annealing oligos, which in turn depends on the amount of template (including PCR product) in a given cycle. This concentration increases a great deal during a PCR; fortunately PCR seems quite robust for a variety of oligo melting temperatures.

Max Ns Accepted

Maximum number of unknown bases (N) allowable in any primer.

Liberal Base

This parameter provides a quick-and-dirty way to get Primer3 to accept IUB / IUPAC codes for ambiguous bases (i.e. by changing all unrecognized bases to N). If you wish to include an ambiguous base in an oligo, you must set Max Ns Accepted to a non-0 value. Perhaps '-' and '* ' should be squeezed out rather than changed to 'N', but currently they simply get converted to N's. The authors invite user comments.

First Base Index

The index of the first base in the input sequence. For input and output using 1-based indexing (such as that used in GenBank and to which many users are accustomed) set this parameter to 1. For input and output using 0-based indexing set this parameter to 0. (This parameter also affects the indexes in the contents of the files produced when the primer file flag is set.) In the WWW interface this parameter defaults to 1.

Inside Target Penalty

Non-default values valid only for sequences with 0 or 1 target regions. If the primer is part of a pair that spans a target and overlaps the target, then multiply this value times the number of nucleotide positions by which the primer overlaps the (unique) target to get the 'position penalty'. The effect of this parameter is to allow Primer3 to include overlap with the target as a term in the objective function.

Outside Target Penalty

Non-default values valid only for sequences with 0 or 1 target regions. If the primer is part of a pair that spans a target and does not overlap the target, then multiply this value times the number of nucleotide positions from the 3' end to the (unique) target to get the 'position penalty'. The effect of this parameter is to allow Primer3 to include nearness to the target as a term in the objective function.

Show Debuging Info

Include the input to primer3_core as part of the output.

Sequence Quality

Sequence Quality

A list of space separated integers. There must be exactly one integer for each base in the Source Sequence if this argument is non-empty. High numbers indicate high confidence in the base call at that position and low numbers indicate low confidence in the base call at that position.

Min Sequence Quality

The minimum sequence quality (as specified by Sequence Quality) allowed within a primer.

Min 3' Sequence Quality

The minimum sequence quality (as specified by Sequence Quality) allowed within the 3' pentamer of a primer.

Sequence Quality Range Min

The minimum legal sequence quality (used for interpreting Min Sequence Quality and Min 3' Sequence Quality).

Sequence Quality Range Max

The maximum legal sequence quality (used for interpreting Min Sequence Quality and Min 3' Sequence Quality).

Penalty Weights

This section describes "penalty weights", which allow the user to modify the criteria that Primer3 uses to select the "best" primers. There are two classes of weights: for some parameters there is a 'Lt' (less than) and a 'Gt' (greater than) weight. These are the weights that Primer3 uses when the value is less or greater than (respectively) the specified optimum. The following parameters have both 'Lt' and 'Gt' weights:

- Product Size
- Primer Size
- Primer T_m
- Product T_m
- Primer GC%
- Hyb Oligo Size
- Hyb Oligo T_m
- Hyb Oligo GC%

The Inside Target Penalty and Outside Target Penalty are similar, except that since they relate to position they do not lend them selves to the 'Lt' and 'Gt' nomenclature.

For the remaining parameters the optimum is understood and the actual value can only vary in one direction from the optimum:

- Primer Self Complementarity
- Primer 3' Self Complementarity
- Primer #N's
- Primer Mispriming Similarity
- Primer Sequence Quality
- Primer 3' Sequence Quality
- Primer 3' Stability
- Hyb Oligo Self Complementarity
- Hyb Oligo 3' Self Complementarity
- Hyb Oligo Mispriming Similarity
- Hyb Oligo Sequence Quality
- Hyb Oligo 3' Sequence Quality

The following are weights are treated specially:

Position Penalty Weight

Determines the overall weight of the position penalty in calculating the penalty for a primer.

Primer Weight

Determines the weight of the 2 primer penalties in calculating the primer pair penalty.

Hyb Oligo Weight

Determines the weight of the hyb oligo penalty in calculating the penalty of a primer pair plus hyb oligo.

The following govern the weight given to various parameters of primer pairs (or primer pairs plus hyb oligo).

- T_m difference
- Primer-Primer Complementarity
- Primer-Primer 3' Complementarity
- Primer Pair Mispriming Similarity

Hyb Oligos (Internal Oligos)

Parameters governing choice of internal oligos are analogous to the parameters governing choice of primer pairs. The exception is Max 3' Complementarity which is meaningless when applied to internal oligos used for hybridization-based detection, since primer-dimer will not occur. We recommend that Max 3' Complementarity be set at least as high as Max Complementarity.

Copyright Notice and Disclaimer

We also request that use of this software be cited in publications as
Steve Rozen and Helen J. Skaletsky (2000) Primer3 on the WWW for general users and for biologist programmers. In: Krawetz S, Misener S (eds) *Bioinformatics Methods and Protocols: Methods in Molecular Biology*. Humana Press, Totowa, NJ, pp 365-386
Source code available at http://fokker.wi.mit.edu/primer3/.

Acknowledgments

The development of Primer3 and the Primer3 web site was funded by Howard Hughes Medical Institute and by the National Institutes of Health, National Human Genome Research Institute. under grants R01-HG00257 (to David C. Page) and P50-HG00098 (to Eric S. Lander).
We gratefully acknowledge the support of Digital Equipment Corporation, which provided the Alphas which were used for much of the development of Primer3, and of Centerline Software, Inc., whose TestCenter memory-error, -leak, and test-coverage checker we use regularly to discover and correct otherwise latent errors in Primer3.

Web software provided by Steve Rozen and Whitehead Institute for Biomedical Research.

Parameters:

Input	
Input file	Sequence Database must already be formatted by formatdb.
PRIMER_MISPRIMING_LIBRARY	The name of a file containing a nucleotide sequence library of sequences to avoid amplifying (for example repetitive sequences, or possibly the sequences of genes in a gene family that should not be amplified.) The file must be in FASTA format.
Output	
Result	Name of the output file
Options	
TARGET	if one or more Targets is specified then a legal primer pair must flank at least one of them. A Target might be a simple sequence repeat site (for example a CA repeat) or a single-base-pair polymorphism. The value should be a space-separated list of

	<start>,<length> pairs where <start> is the index of the first base of a Target, and <length> is its length. For backward compatibility Primer3 accepts (but ignores) a trailing ,<description> for each element of this argument.
EXCLUDED_REGION	Primer oligos may not overlap any region specified in this tag. The associated value must be a space-separated list of <start>,<length> pairs where <start> is the index of the first base of the excluded region, and <length> is its length. This tag is useful for tasks such as excluding regions of low sequence quality or for excluding regions containing repetitive elements such as ALUs or LINEs.
PRIMER_SEQUENCE_QUALITY	A list of space separated integers. There must be exactly one integer for each base in input sequence if this argument is non-empty. For example, for the sequence ANNTTCAG... PRIMER_SEQUENCE_QUALITY might be 45 10 0 50 30 34 50 67 .... High numbers indicate high confidence in the base called at that position and low numbers indicate low confidence in the base call at that position. This parameter is only relevant if you are using a base calling program that provides quality information (for example phred).
PRIMER_LEFT_INPUT	The sequence of a left primer to check and around which to design right primers and optional internal oligos. Must be a substring of an input sequence.
PRIMER_RIGHT_INPUT	The sequence of a right primer to check and around which to design left primers and optional internal oligos. Must be a substring of the reverse strand of an input sequence.
PRIMER_START_CODON_POSITION	This parameter should be considered EXPERIMENTAL at this point. Please check the output carefully; some erroneous inputs might cause an error in Primer3. Index of the first base of a start codon. This parameter allows Primer3 to select primer pairs to create in-frame amplicons e.g. to create a template for a fusion protein. Primer3 will attempt to select an in-frame left primer, ideally starting at or to the left of the start codon, or to the right if necessary. Negative values of this parameter are legal if the actual start codon is to the left of available sequence. If this parameter is non-negative Primer3 signals an error if the codon at the position specified by this parameter is not an ATG. A value less than or equal to -10^6 indicates that Primer3 should ignore this parameter. Primer3 selects the position of the right primer by scanning right from the left primer for a stop codon. Ideally the right primer will end at or after the stop codon.
PRIMER_PICK_ANYWAY	If true pick a primer pair even if PRIMER_LEFT_INPUT, PRIMER_RIGHT_INPUT, or PRIMER_INTERNAL_OLIGO_INPUT violates specific constraints.
PRIMER_LIB_AMBIGUITY_CODES_CONSENSUS	If set to 1, treat ambiguity codes as if they were consensus codes when matching oligos to mispriming or mishyb libraries. For example, if this flag is set, then a C in an oligo will be scored as a perfect match to an S in a library sequence, as will a G in the

	oligo. More importantly, though, any base in an oligo will be scored as a perfect match to an N in the library. This is very bad if the library contains strings of Ns, as no oligo will be legal (and it will take a long time to find this out). So unless you know for sure that your library does not have runs of Ns (or Xs), then set this flag to 0.
PRIMER_MAX_MISPRIMING	The maximum allowed weighted similarity with any sequence in PRIMER_MISPRIMING_LIBRARY.
PRIMER_MAX_TEMPLATE_ MISPRIMING	The maximum allowed similarity to ectopic sites in the template. A negative value means do not check. The scoring system is the same as used for PRIMER_MAX_MISPRIMING, except that an ambiguity code in the template is never treated as a consensus (see PRIMER_LIB_AMBIGUITY_CODES_CONSENSUS).
PRIMER_PAIR_MAX_MISPRIMING	The maximum allowed sum of similarities of a primer pair (one similarity for each primer) with any single sequence in PRIMER_MISPRIMING_LIBRARY. Library sequence weights are not used in computing the sum of similarities.
PRIMER_PAIR_MAX_TEMPLATE_MISPRIMING	The maximum allowed summed similarity of both primers to ectopic sites in the template. A negative value means do not check. The scoring system is the same as used for PRIMER_PAIR_MAX_MISPRIMING, except that an ambiguity code in the template is never treated as a consensus (see PRIMER_LIB_AMBIGUITY_CODES_CONSENSUS). Primer3 does not check the similarity of hybridization oligos (internal oligos) to locations outside of the amplicon.
PRIMER_PRODUCT_MAX_TM	The maximum allowed melting temperature of the amplicon. Primer3 calculates product Tm calculated using the formula from Bolton and McCarthy, PNAS 84:1390 (1962) as presented in Sambrook, Fritsch and Maniatis, Molecular Cloning, p 11.46 (1989, CSHL Press). Tm = 81.5 + 16.6(log10([Na+])) + .41*(%GC) - 600/length Where [Na+] is the molar sodium concentration, (%GC) is the percent of Gs and Cs in the sequence, and length is the length of the sequence. A similar formula is used by the prime primer selection program in GCG (http://www.gcg.com), which instead uses 675.0 / length in the last term (after F. Baldino, Jr, M.-F. Chesselet, and M.E. Lewis, Methods in Enzymology 168:766 (1989) eqn (1) on page 766 without the mismatch and formamide terms). The formulas here and in Baldino et al. assume Na+ rather than K+. According to J.G. Wetmur, Critical Reviews in BioChem. and Mol. Bio. 26:227 (1991) 50 mM K+ should be equivalent in these formulae to .2 M Na+. Primer3 uses the same salt concentration value for calculating both the primer melting temperature and the oligo melting temperature. If you are planning to use the PCR product for hybridization later this behavior will not give you the Tm under hybridization conditions.
PRIMER_PRODUCT_MIN_TM	The minimum allowed melting temperature of the amplicon. Please see the documentation on the maximum melting temperature of the product for details.
PRIMER_EXPLAIN_FLAG	If this flag is non-0, produce PRIMER_LEFT_EXPLAIN,

	PRIMER_RIGHT_EXPLAIN, and PRIMER_INTERNAL_OLIGO_EXPLAIN output tags, which are intended to provide information on the number of oligos and primer pairs that Primer3 examined, and statistics on the number discarded for various reasons. If format_output is set similar information is produced in the user-oriented output.
PRIMER_PRODUCT_SIZE_RANGE	The associated values specify the lengths of the product that the user wants the primers to create, and is a space separated list of elements of the form <x>-<y> where an <x>-<y> pair is a legal range of lengths for the product. For example, if one wants PCR products to be between 100 to 150 bases (inclusive) then one would set this parameter to 100-150. If one desires PCR products in either the range from 100 to 150 bases or in the range from 200 to 250 bases then one would set this parameter to 100-150 200-250. Primer3 favors ranges to the left side of the parameter string. Primer3 will return legal primers pairs in the first range regardless the value of the objective function for these pairs. Only if there are an insufficient number of primers in the first range will Primer3 return primers in a subsequent range.
PRIMER_PICK_INTERNAL_OLIGO	If the associated value is non-0, then Primer3 will attempt to pick an internal oligo (hybridization probe to detect the PCR product). This tag is maintained for backward compatibility. Use PRIMER_TASK.
PRIMER_GC_CLAMP	Require the specified number of consecutive Gs and Cs at the 3' end of both the left and right primer. (This parameter has no effect on the internal oligo if one is requested.)
PRIMER_OPT_SIZE	Optimum length (in bases) of a primer oligo. Primer3 will attempt to pick primers close to this length.
PRIMER_DEFAULT_SIZE	A deprecated synonym for PRIMER_OPT_SIZE, maintained for v2 compatibility.
PRIMER_MIN_SIZE	Minimum acceptable length of a primer. Must be greater than 0 and less than or equal to PRIMER_MAX_SIZE.
PRIMER_MAX_SIZE	Maximum acceptable length (in bases) of a primer. Currently this parameter cannot be larger than 35. This limit is governed by maximum oligo size for which Primer3's melting-temperature is valid.
PRIMER_OPT_TM	Optimum melting temperature(Celsius) for a primer oligo. Primer3 will try to pick primers with melting temperatures are close to this temperature. The oligo melting temperature formula in Primer3 is that given in Rychlik, Spencer and Rhoads, Nucleic Acids Research, 18(21): 6409-6412 and Breslauer, Frank, Bloeker and Marky, PNAS, 83: 3746-3750. Please refer to the former paper for background discussion.
PRIMER_MIN_TM	Minimum acceptable melting temperature(Celsius) for a primer oligo.
PRIMER_MAX_TM	Maximum acceptable melting temperature(Celsius) for a primer oligo.
PRIMER_MAX_DIFF_TM	Maximum acceptable (unsigned) difference between the melting temperatures of the left and right primers.

PRIMER_MIN_GC	Minimum allowable percentage of Gs and Cs in any primer.
PRIMER_OPT_GC_PERCENT	Optimum GC percent. This parameter influences primer selection only if PRIMER_WT_GC_PERCENT_GT or PRIMER_WT_GC_PERCENT_LT are non-0.
PRIMER_MAX_GC	Maximum allowable percentage of Gs and Cs in any primer generated by Primer.
PRIMER_SALT_CONC	The millimolar concentration of salt (usually KCl) in the PCR. Primer3 uses this argument to calculate oligo melting temperatures.
PRIMER_DNA_CONC	The nanomolar concentration of annealing oligos in the PCR. Primer3 uses this argument to calculate oligo melting temperatures. The default (50nM) works well with the standard protocol used at the Whitehead/MIT Center for Genome Research--0.5 microliters of 20 micromolar concentration for each primer oligo in a 20 microliter reaction with 10 nanograms template, 0.025 units/microliter Taq polymerase in 0.1 mM each dNTP, 1.5mM MgCl2, 50mM KCl, 10mM Tris-HCL (pH 9.3) using 35 cycles with an annealing temperature of 56 degrees Celsius. This parameter corresponds to 'c' in Rychlik, Spencer and Rhoads' equation (ii) (Nucleic Acids Research, 18(21): 6409-6412) where a suitable value (for a lower initial concentration of template) is "empirically determined". The value of this parameter is less than the actual concentration of oligos in the reaction because it is the concentration of annealing oligos, which in turn depends on the amount of template (including PCR product) in a given cycle. This concentration increases a great deal during a PCR; fortunately PCR seems quite robust for a variety of oligo melting temperatures.
PRIMER_NUM_NS_ACCEPTED	Maximum number of unknown bases (N) allowable in any primer.

PRIMER_SELF_ANY	The maximum allowable local alignment score when testing a single primer for (local) self-complementarity and the maximum allowable local alignment score when testing for complementarity between left and right primers. Local self-complementarity is taken to predict the tendency of primers to anneal to each other without necessarily causing self-priming in the PCR. The scoring system gives 1.00 for complementary bases, -0.25 for a match of any base (or N) with an N, -1.00 for a mismatch, and -2.00 for a gap. Only single-base-pair gaps are allowed. For example, the alignment
	`5' ATCGNA 3'`
	`   || | |`
	`3' TA-CGT 5'`
	is allowed (and yields a score of 1.75), but the alignment
	`5' ATCCGNA 3'`
	`   ||  | |`
	`3' TA--CGT 5'`
	is not considered. Scores are non-negative, and a score of 0.00 indicates that there is no reasonable local alignment between two oligos.

PRIMER_SELF_END	The maximum allowable 3'-anchored global alignment score when testing a single primer for self-complementarity, and the

<table>
<tr><td></td><td>maximum allowable 3'-anchored global alignment score when testing for complementarity between left and right primers. The 3'-anchored global alignment score is taken to predict the likelihood of PCR-priming primer-dimers, for example
 5' ATGCCCTAGCTTCCGGATG 3'
              ||| |||||
           3' AAGTCCTACATTTAGCCTAGT 5'
or
 5` AGGCTATGGGCCTCGCGA 3'
                ||||||
             3' AGCGCTCCGGGTATCGGA 5'
The scoring system is as for the Maximum Complementarity argument. In the examples above the scores are 7.00 and 6.00 respectively. Scores are non-negative, and a score of 0.00 indicates that there is no reasonable 3'-anchored global alignment between two oligos. In order to estimate 3'-anchored global alignments for candidate primers and primer pairs, Primer assumes that the sequence from which to choose primers is presented 5'->3'. It is nonsensical to provide a larger value for this parameter than for the Maximum (local) Complementarity parameter because the score of a local alignment will always be at least as great as the score of a global alignment.</td></tr>
<tr><td><b>PRIMER_MAX_POLY_X</b></td><td>The maximum allowable length of a mononucleotide repeat, for example AAAAAA.</td></tr>
<tr><td><b>PRIMER_LIBERAL_BASE</b></td><td>This parameter provides a quick-and-dirty way to get Primer3 to accept IUB / IUPAC codes for ambiguous bases (i.e. by changing all unrecognized bases to N). If you wish to include an ambiguous base in an oligo, you must set PRIMER_NUM_NS_ACCEPTED to a non-0 value.
Perhaps '-' and '* ' should be squeezed out rather than changed to 'N', but currently they simply get converted to N's. The authors invite user comments.</td></tr>
<tr><td><b>PRIMER_NUM_RETURN</b></td><td>The maximum number of primer pairs to return. Primer pairs returned are sorted by their "quality", in other words by the value of the objective function (where a lower number indicates a better primer pair). Caution: setting this parameter to a large value will increase running time.</td></tr>
<tr><td><b>PRIMER_FIRST_BASE_INDEX</b></td><td>This parameter is the index of the first base in the input sequence. For input and output using 1-based indexing (such as that used in GenBank and to which many users are accustomed) set this parameter to 1. For input and output using 0-based indexing set this parameter to 0. (This parameter also affects the indexes in the contents of the files produced when the primer file flag is set.)</td></tr>
<tr><td><b>PRIMER_MIN_QUALITY</b></td><td>The minimum sequence quality (as specified by PRIMER_SEQUENCE_QUALITY) allowed within a primer.</td></tr>
<tr><td><b>PRIMER_MIN_END_QUALITY</b></td><td>The minimum sequence quality (as specified by PRIMER_SEQUENCE_QUALITY) allowed within the 5' pentamer of a primer.</td></tr>
<tr><td><b>PRIMER_QUALITY_RANGE_MIN</b></td><td>The minimum legal sequence quality (used for error checking of PRIMER_MIN_QUALITY and PRIMER_MIN_END_QUALITY).</td></tr>
<tr><td><b>PRIMER_INSIDE_PENALTY</b></td><td>This experimental parameter might not be maintained in this form</td></tr>
</table>

	in the next release. Non-default values valid only for sequences with 0 or 1 target regions. If the primer is part of a pair that spans a target and overlaps the target, then multiply this value times the number of nucleotide positions by which the primer overlaps the (unique) target to get the 'position penalty'. The effect of this parameter is to allow Primer3 to include overlap with the target as a term in the objective function.
PRIMER_OUTSIDE_PENALTY	This experimental parameter might not be maintained in this form in the next release. Non-default values valid only for sequences with 0 or 1 target regions. If the primer is part of a pair that spans a target and does not overlap the target, then multiply this value times the number of nucleotide positions from the 3' end to the (unique) target to get the 'position penalty'. The effect of this parameter is to allow Primer3 to include nearness to the target as a term in the objective function.
PRIMER_MAX_END_STABILITY	The maximum stability for the five 3' bases of a left or right primer. Bigger numbers mean more stable 3' ends. The value is the maximum delta G for duplex disruption for the five 3' bases as calculated using the nearest neighbor parameters published in Breslauer, Frank, Bloeker and Marky, Proc. Natl. Acad. Sci. USA, vol 83, pp 3746-3750. Primer3 uses a completely permissive default value for backward compatibility (which we may change in the next release). Rychlik recommends a maximum value of 9 (Wojciech Rychlik, "Selection of Primers for Polymerase Chain Reaction" in BA White, Ed., "Methods in Molecular Biology, Vol. 15: PCR Protocols: Current Methods and Applications", 1993, pp 31-40, Humana Press, Totowa NJ).
PRIMER_PRODUCT_OPT_TM	The optimum melting temperature for the PCR product. 0 indicates that there is no optimum temperature.
PRIMER_PRODUCT_OPT_SIZE	The optimum size for the PCR product. 0 indicates that there is no optimum product size. This parameter influences primer pair selection only if PRIMER_PAIR_WT_PRODUCT_SIZE_GT or PRIMER_PAIR_WT_PRODUCT_SIZE_LT is non-0.
PRIMER_TASK	Tell Primer3 what task to perform. The tasks should be self explanatory, except that we note that pick_pcr_primers_and_hyb_probe is equivalent to the setting PRIMER_PICK_INTERNAL_OLIGO to a non-zero value and setting PRIMER_TASK to pick_pcr_primers.
pick_pcr_primers	PRIMER_TASK
pick_pcr_primers_and_hyb_probe	PRIMER_TASK
pick_left_only	PRIMER_TASK
pick_right_only	PRIMER_TASK
pick_hyb_probe_only	PRIMER_TASK
PRIMER_WT_TM_GT	Penalty weight for primers with Tm over PRIMER_OPT_TM.
PRIMER_WT_TM_LT	Penalty weight for primers with Tm under PRIMER_OPT_TM.
PRIMER_WT_SIZE_LT	Penalty weight for primers shorter than PRIMER_OPT_SIZE.
PRIMER_WT_SIZE_GT	Penalty weight for primers longer than PRIMER_OPT_SIZE.
PRIMER_WT_GC_PERCENT_	Penalty weight for primers with GC percent greater than

LT	PRIMER_OPT_GC_PERCENT.
PRIMER_WT_GC_PERCENT_GT	Penalty weight for primers with GC percent greater than PRIMER_OPT_GC_PERCENT.
PRIMER_INTERNAL_OLIGO_EXCLUDED_REGION	Middle oligos may not overlap any region specified by this tag. The associated value must be a space-separated list of <start>,<length> pairs, where <start> is the index of the first base of an excluded region, and <length> is its length. Often one would make Target regions excluded regions for internal oligos.
PRIMER_INTERNAL_OLIGO_INPUT	The sequence of an internal oligo to check and around which to design left and right primers. Must be a substring of SEQUENCE.
PRIMER_INTERNAL_OLIGO_MISHYB_LIBRARY	Similar to PRIMER_MISPRIMING_LIBRARY, except that the event we seek to avoid is hybridization of the internal oligo to sequences in this library rather than priming from them.
PRIMER_INTERNAL_OLIGO_MAX_MISHYB	Similar to PRIMER_MAX_MISPRIMING except that this parameter applies to the similarity of candidate internal oligos to the library specified in PRIMER_INTERNAL_OLIGO_MISHYB_LIBRARY.
PRIMER_INTERNAL_OLIGO_MIN_QUALITY	(Note that there is no PRIMER_INTERNAL_OLIGO_MIN_END_QUALITY.)

ReplaceSeq

ReplaceSeq is a procedure for replacing of a given string with another string in a file.

Parameters:

	Input
Target sequence	Name of the input file
	Output
Result	Name of the output file
	Options
String to search	String to search
To replace with	To replace with

Restrictase

The program for finding and displaying the positions of the cut sites of restriction enzyme recognition sequences. This program displays the cut sites on both strands by default. This program uses The Restriction Enzyme database (REBASE). The home page of REBASE is: http://rebase.neb.com/

Description of REBASE, The Restriction Enzyme Database

```
    REBASE, The Restriction Enzyme Database    http://rebase.neb.com
    Copyright (c)  Dr. Richard J. Roberts, 2006.   All rights reserved.

                USER MANUAL FOR REBASE's 'bairoch' FORMAT


1. INTRODUCTION

The file bairoch.### contains  an alphabetical listing of type I, II and III
restriction enzymes as well as methylases in a format compatible with that of
```

the EMBL, SWISS-PROT, ENZYME, PROSITE, ECD, EPD, and HAEMB data banks. It can also be used with PC/Gene.

Each entry is composed of lines. Different types of lines, each with their own format, are used to record the various data which make up the entry. A sample entry is shown here:

```
        ID   AluI
        AC   RB30
        ET   R2 M
        OS   Arthrobacter luteus
        PT   AluI
        RS   AGCT, 2;
        MS   3(5mC);
        CR   A,B,E,F,H,I,K,L,M,N,O,P,Q,R,S,U,V,X.
        CM   A,E,K,N,U.
        RN   [1]
        RA   Kramarov V.M., Smolyaninov V.V.;
        RL   Biokhimiya 46:1526-1529(1981).
        RN   [2]
        RA   Roberts R.J., Myers P.A., Morrison A., Murray K.;
        RL   J. Mol. Biol. 102:157-165(1976).
        RN   [3]
        RA   Yoon H., Suh H., Han M.H., Yoo O.J.;
        RL   Korean Biochem. J. 18:82-87(1985).
        RN   [4]
        RA   Yoon H., Suh H., Kim K., Han M.H., Yoo O.J.;
        RL   Korean Biochem. J. 18:88-93(1985).
        //
```

Each line begins with a two-character line code, which indicates the type of data contained in the line. The current line types and line codes and the order in which they appear in an entry, are shown below:

```
        ID     - Enzyme acronym
        AC     - REBASE accession number
        ET     - Enzyme type
        OS     - Organism species
        PT     - Prototype
        RS     - Recognition sequence(s), cut site(s)
        MS     - Methylation site(s) and type               [optional]
        CR     - Commercial sources for the restriction enzyme   [optional]
        CM     - Commercial sources for the methylase       [optional]
        RN     - Reference number
        RA     - Reference authors
        RL     - Reference location
        //     - Termination line
```

2. THE DIFFERENT LINE TYPES

2.1 The ID line.

The ID (IDentification) line is always the first line of an entry and shows the restriction enzyme acronym or the methylase acronym if no corresponding restriction enzyme with this acronym exists. Examples:

```
        ID   EcoRI
        ID   Sau3AI
        ID   M.NgoVIII
```

2.2 The ET line.

The ET (Enzyme Type) line shows what type(s) of enzyme are described in an entry. The following codes are used:

```
    Rn   : where `n' is the type of the restriction enzyme (from 1 to 3).
    M    : indicates that there is a corresponding methylase.
    Rn*  : indicates the restriction enzyme is of type n, but only recognizes
           the sequence when it is methylated.
    IE   : indicates that this is an intron-encoded (homing) endonuclease
```

Example:

```
        ET   R2 M
```

Describes a type-II restriction enzyme (R2) and the corresponding methylase (M).

2.3 The OS line.

The OS (Organism Species) line specifies the organism which was the source of the stored enzymes. In the current version strain information is included in the OS line. Examples:

```
        OS   Escherichia coli RY13
        OS   Neisseria meningitidis DRES-30
```

2.4 The PT line.

The PT (Prototype) line specifies the acronym of the prototype enzyme.

2.5 The RS line.

The RS (Recognition Sequence(s), cut site(s)) line follows the syntax:

```
        RS   site1, cut1; [site2, cut2];
```

Where siteN is a recognition site, and cutN the offset in bases of the cleavage site from the beginning of the recognition site. Examples:

```
        RS   CAGCAC, 0;
        RS   CAGCAC, 1;
```

In the first case shown above the enzyme cleaves before the first base of the recognition site (offset=0; ^CAGCAC), while in the second case it cuts between the first and second bases (offset=1; C^AGCAC).

If the recognition site or the cleavage site are unknown a question mark is used. Examples:

```
        RS   CAGCAC, ?;
        RS   ?, ?;
```

For asymmetric restriction enzyme (non palindromic) the two recognition sites are indicated. Example for FokI:

```
        RS   GGATG, 14; CATCC, -13;
```

2.6 The MS line.

The MS (Methylation Site(s) and type) line follows the format:

```
        MS   b1(t1)[,b2(t2)];
```

Where b1 and b2 are numbers that refer to the position of the 3'methylated and 5'methylated bases (the numbering system starts at 1 with the first base of the recognition sequence and is negative if the base is upstream of the recognition sequence)

Where t1 and t2 are acronyms that indicate the type of methylation which can be one of the following:

```
        N4mC = N4-methylcytosine
        5mC  = 5-methylcytosine
        6mA  = 6-methyladenosine.
```

Examples:

```
        MS   5(N4mC);
```

Indicates a N4-methylcytosine on base 5.

```
        MS   3(6mA),-2(6mA);
```

Indicates a 6-methylcytosine on the 3'base 3 and on the 5'base -2.

If the methylation site is unknown a question mark is used. Example:

```
        MS   ?(6mA);
```

The MS line is optional: it does not appear in an entry if there are no known methylase associated with the restriction enzyme being described by that entry.

2.7 The CR and CM lines.

The CR and CM lines are used to show the commercial sources of restriction enzymes (CR) and of methylases (CM). The format of these line is:

```
        CR   A1[,A2,A3,...,An].
```

Where A1 to An are abbreviations for commercial suppliers. At the end of this file, is a complete list of the abbreviations currently defined in REBASE, in the following format:

```
                N        New England Biolabs (11/05)
                R        Promega Corporation (9/05)
```

(the date within the parentheses indicates the last update to each suppliers listing in REBASE)

Examples:

```
        CR   A,B,E,I,J,K,L,M,N,O,P,Q,R,S,U,V,X.
        CM   A,E,K,N,U.
```

The CR and CM lines are optional: they do not appear in an entry if an enzyme or a methylase are not available from any of the commercial companies listed above.

2.8 The references lines (RN, RA, and RL).

These lines comprise the literature citations within REBASE. The citations indicate the papers from which the data has been abstracted. The reference lines for a given citation occur in a block, and are always in the order RN, RA, RL. Within each such reference block the RN and RL lines occur once, while the RA line occurs one or more times. If several references are given, there will be a reference block for each.

An example of a complete reference is:

```
RN   [1]
RA   Gelinas R.E., Myers P.A., Weiss G.H., Roberts R.J., Murray K.;
RL   J. Mol. Biol. 114:433-440(1977).
```

2.8.1 The RN line

The RN (Reference Number) line gives a sequential number to each reference citation in an entry. The format of the RN line is:

```
RN   [N]
```

where `N' denotes the nth reference for this entry. The reference number is always enclosed in square brackets.

2.8.2 The RA line

The RA (Reference Author) lines list the authors of the paper (or other work) cited. All of the authors are included, and are listed in the order given in the paper. The names are listed surname first followed by a blank followed by initial(s) with periods. The authors' names are separated by commas and terminated by a semicolon. Author names are not split between lines. An example of the use of RA lines is shown below:

```
RA   Gelinas R.E., Myers P.A., Weiss G.H., Roberts R.J., Murray K.;
```

2.8.3 The RL line

The RL (Reference Location) line contains the citation information for the reference. The RL line for a journal citation includes the journal abbreviation, the volume number, the page range, and the year. The format for such a RL line is:

```
RL   JOURNAL VOL:PP-PP(YEAR).
```

RL lines for unpublished results follows the format shown in the following example:

```
RL   Unpublished observations.
```

2.9 The // line.

The // (terminator) line contains no data or comments. It designates the end of an entry.

2.10 CC lines.

Any line beginning with CC will be treated as a comment.

+ +

Table 1. Summary of single-letter code recommendations

Symbol	Meaning	Origin of designation
G	G	Guanine
A	A	Adenine
T	T	Thymine
C	C	Cytosine
R	G or A	puRine
Y	T or C	pYrimidine
M	A or C	aMino

K	G or T	Keto
S	G or C	Strong interaction (3 H bonds)
W	A or T	Weak interaction (2 H bonds)
H	A or C or T	not-G, H follows G in the alphabet
B	G or T or C	not-A, B follows A
V	G or C or A	not-T (not-U), V follows U
D	G or A or T	not-C, D follows C
N	G or A or T or C	aNy

Output example

```
Kpn49kI
Uba58I
RsrI
SsoI
M.CjeNI
M.RsrI
M.SsoI
VchO2I
Srl55DI
Eco159I
Eco228I
HalI
FunII
VchN100I
Hal22I
Ppu111I
Srl32DII
Eco252I
M.Ppu111I
Van91II
M.EcoRI
M.Van91II
Eco237I
Eco82I
EcoRI
|
Gaattctaatctccctctcaaccctacagtcacccatttggtatattaaagatgtgttgt
         10        20        30        40        50
CttaagattagagggagagttgggatgtcagtgggtaaaccatataatttctaCacaaca
|                                                    |
EcoRI                                                BsbI
Eco82I
Eco237I
M.Van91II
M.EcoRI
Van91II
M.Ppu111I
Eco252I
Srl32DII
Ppu111I
Hal22I
VchN100I
FunII
HalI
Eco228I
Eco159I
Srl55DI
VchO2I
M.SsoI
M.RsrI
M.CjeNI
SsoI
RsrI
Uba58I
Kpn49kI
```

```
                                          MspSWI
```

```
                                        BstRZ246I
                                        BstSWI
                                        M.SwaI
                                        SwaI
                                        SmiI
                                        |DraI
                                        |M.DraI
                                        |AhaIII
                                        |PauAII
                                        |M.EsaDix1I
                                        |SruI
                                        |Srl76DI
          BfuI                          |Srl19I
          BciVI                         |Srl61DI
          |                             ||
ctactgtctaGtatccctcaagtagtgtcaggaattagtcATttaaatagtctgcaagcc
         70        80        90        100       110
gatgacagatcataggGagttcatcacagtccttaatcagTAaatttatcagacgttcgg
                |                       ||
                Bce83I                  |Srl61DI
                BpuEI                   |Srl19I
                                        |Srl76DI
                                        |SruI
                                        |M.EsaDix1I
                                        |PauAII
                                        |AhaIII
                                        |M.DraI
                                        |DraI
                                        SmiI
                                        SwaI
                                        M.SwaI
                                        BstSWI
                                        BstRZ246I
                                        MspSWI

                                BpmI
                                Bco35I
                                BspJ74I
                                M.GsuI
                                M.BpmI
                                Bsp22I
                                Uba1444I
                                GsuI
                                Bsp28I
                                Bth1795I                  BpuEI
                                Uba1437I                  Bce83I
                                |                         |
aggagtggtggctcatgtctgtaattccagcaCtggagaggtagaagtgggaggactgCt
         130       140       150       160       170
tcctcaccaccgagtacagacattaaggtcgtGacctctccatcttcaccctcctgacga
                                |
                                M.BpmI
                                M.GsuI

 ScoI
 Psp124BI
 SacI
 Ecl136II
 EcoICRI
 M.SstI
 Eco53kI
 SstI
 NasSI
 MxaI
 M.SacI
 Pfl18I
 Ecl137I
 BpuAmI                 BspGI
 |                      |
tGagctcaagagtttgatattatcCtggac
         190       200       210
```

```
aCtcGagttctcaaactataataggacctg
 |  |
 |  Bce83I
 |  BpuEI
 BpuAmI
 Ecl137I
 Pfl18I
 M.SacI
 MxaI
 NasSI
 SstI
 Eco53kI
 M.SstI
 EcoICRI
 Ecl136II
 SacI
 Psp124BI
 ScoI

Commercially Available (total 15):
Enzyme        Direct                         Reverse
name          chain                          chain
--------------------------------------------------------------------------------
BciVI         GTATCC                         GGATAC
BfuI          GTATCC                         GGATAC
BpmI          CTGGAG                         CTCCAG
BpuEI         CTTGAG                         CTCAAG
DraI          TTTAAA                         TTTAAA
Ecl136II      GAGCTC                         GAGCTC
EcoICRI       GAGCTC                         GAGCTC
EcoRI         GAATTC                         GAATTC
GsuI          CTGGAG                         CTCCAG
M.EcoRI       GAATTC                         GAATTC
Psp124BI      GAGCTC                         GAGCTC
SacI          GAGCTC                         GAGCTC
SmiI          ATTTAAAT                       ATTTAAAT
SstI          GAGCTC                         GAGCTC
SwaI          ATTTAAAT                       ATTTAAAT

In direct chain (total 70):
Enzyme        Recognition                    Cut  No.  Positions
name          sequence                       site cuts of sites
--------------------------------------------------------------------------------
AhaIII        TTT^AAA                        3    1    102
Bce83I        CTTGAG                         22   1    179
BciVI         GTATCC                         12   1    71
Bco35I        CTGGAG                         ?    1    153
BfuI          GTATCC                         12   1    71
BpmI          CTGGAG                         22   1    153
BpuAmI        GAG^CTC                        3    1    182
BpuEI         CTTGAG                         22   1    179
Bsp22I        CTGGAG                         ?    1    153
Bsp28I        CTGGAG                         ?    1    153
BspGI         CTGGAC                         ?    1    205
BspJ74I       CTGGAG                         ?    1    153
BstRZ246I     ATTT^AAAT                      4    1    101
BstSWI        ATTT^AAAT                      4    1    101
Bth1795I      CTGGAG                         ?    1    153
DraI          TTT^AAA                        3    1    102
Ecl136II      GAG^CTC                        3    1    182
Ecl137I       GAGCTC                         ?    1    182
Eco159I       GAATTC                         ?    1    1
Eco228I       GAATTC                         ?    1    1
Eco237I       GAATTC                         ?    1    1
Eco252I       GAATTC                         ?    1    1
Eco53kI       GAG^CTC                        3    1    182
Eco82I        GAATTC                         ?    1    1
EcoICRI       GAG^CTC                        3    1    182
EcoRI         G^AATTC                        1    1    1
FunII         G^AATTC                        1    1    1
GsuI          CTGGAG                         22   1    153
```

```
Hal22I        GAATTC                         ?    1    1
HalI          G^AATTC                        1    1    1
Kpn49kI       G^AATTC                        1    1    1
M.BpmI        CTGGAG                         ?    1    153
M.CjeNI       GAATTC                         ?    1    1
M.DraI        TTTAAA                         ?    1    102
M.EcoRI       GAATTC                         ?    1    1
M.EsaDix1I    TTTAAA                         ?    1    102
M.GsuI        CTGGAG                         ?    1    153
M.Ppu111I     GAATTC                         ?    1    1
M.RsrI        GAATTC                         ?    1    1
M.SacI        GAGCTC                         ?    1    182
M.SsoI        GAATTC                         ?    1    1
M.SstI        GAGCTC                         ?    1    182
M.SwaI        ATTTAAAT                       ?    1    101
M.Van91II     GAATTC                         ?    1    1
MspSWI        ATTT^AAAT                      4    1    101
MxaI          GAG^CTC                        3    1    182
NasSI         GAGCTC                         ?    1    182
PauAII        TTT^AAA                        3    1    102
Pfl18I        GAGCTC                         ?    1    182
Ppu111I       G^AATTC                        1    1    1
Psp124BI      GAGCT^C                        5    1    182
RsrI          G^AATTC                        1    1    1
SacI          GAGCT^C                        5    1    182
ScoI          GAGCTC                         ?    1    182
SmiI          ATTT^AAAT                      4    1    101
Srl19I        TTTAAA                         ?    1    102
Srl32DII      G^AATTC                        1    1    1
Srl55DI       G^AATTC                        1    1    1
Srl61DI       TTTAAA                         ?    1    102
Srl76DI       TTTAAA                         ?    1    102
SruI          TTT^AAA                        3    1    102
SsoI          G^AATTC                        1    1    1
SstI          GAGCT^C                        5    1    182
SwaI          ATTT^AAAT                      4    1    101
Uba1437I      CTGGAG                         ?    1    153
Uba1444I      CTGGAG                         ?    1    153
Uba58I        GAATTC                         ?    1    1
Van91II       GAATTC                         ?    1    1
VchN100I      GAATTC                         ?    1    1
VchO2I        GAATTC                         ?    1    1

In reverse chain (total 59):
Enzyme        Recognition                    Cut  No.  Positions
name          sequence                       site cuts of sites
--------------------------------------------------------------------------------
AhaIII        TTT^AAA                        3    1    102
Bce83I        CTCAAG                         -14  2    77 185
BpuAmI        GAG^CTC                        3    1    182
BpuEI         CTCAAG                         -14  2    77 185
BsbI          GTGTTG                         ?    1    54
BstRZ246I     ATTT^AAAT                      4    1    101
BstSWI        ATTT^AAAT                      4    1    101
DraI          TTT^AAA                        3    1    102
Ecl136II      GAG^CTC                        3    1    182
Ecl137I       GAGCTC                         ?    1    182
Eco159I       GAATTC                         ?    1    1
Eco228I       GAATTC                         ?    1    1
Eco237I       GAATTC                         ?    1    1
Eco252I       GAATTC                         ?    1    1
Eco53kI       GAG^CTC                        3    1    182
Eco82I        GAATTC                         ?    1    1
EcoICRI       GAG^CTC                        3    1    182
EcoRI         G^AATTC                        1    1    1
FunII         G^AATTC                        1    1    1
Hal22I        GAATTC                         ?    1    1
HalI          G^AATTC                        1    1    1
Kpn49kI       G^AATTC                        1    1    1
M.BpmI        CTGGAG                         ?    1    153
M.CjeNI       GAATTC                         ?    1    1
```

M.DraI	TTTAAA	?	1	102
M.EcoRI	GAATTC	?	1	1
M.EsaDix1I	TTTAAA	?	1	102
M.GsuI	CTGGAG	?	1	153
M.Ppu111I	GAATTC	?	1	1
M.RsrI	GAATTC	?	1	1
M.SacI	GAGCTC	?	1	182
M.SsoI	GAATTC	?	1	1
M.SstI	GAGCTC	?	1	182
M.SwaI	ATTTAAAT	?	1	101
M.Van91II	GAATTC	?	1	1
MspSWI	ATTT^AAAT	4	1	101
MxaI	GAG^CTC	3	1	182
NasSI	GAGCTC	?	1	182
PauAII	TTT^AAA	3	1	102
Pfl18I	GAGCTC	?	1	182
Ppu111I	G^AATTC	1	1	1
Psp124BI	GAGCT^C	5	1	182
RsrI	G^AATTC	1	1	1
SacI	GAGCT^C	5	1	182
ScoI	GAGCTC	?	1	182
SmiI	ATTT^AAAT	4	1	101
Srl19I	TTTAAA	?	1	102
Srl32DII	G^AATTC	1	1	1
Srl55DI	G^AATTC	1	1	1
Srl61DI	TTTAAA	?	1	102
Srl76DI	TTTAAA	?	1	102
SruI	TTT^AAA	3	1	102
SsoI	G^AATTC	1	1	1
SstI	GAGCT^C	5	1	182
SwaI	ATTT^AAAT	4	1	101
Uba58I	GAATTC	?	1	1
Van91II	GAATTC	?	1	1
VchN100I	GAATTC	?	1	1
VchO2I	GAATTC	?	1	1

List of the restrictases from REBASE

Enzyme	Recognition sequence	Recognition sequence	Commercially
name	(direct chain)	(reverse chain)	Available(*)
AaaI	CGGCCG	CGGCCG	
AacI	GGATCC	GGATCC	
M.AacDam	GATC	GATC	
M.Aac465Dam	GATC	GATC	
AaeI	GGATCC	GGATCC	
AagI	ATCGAT	ATCGAT	
AamI	?	?	
AaqI	GTGCAC	GTGCAC	
AarI	CACCTGC	GCAGGTG	F.
AasI	GACNNNNNNGTC	GACNNNNNNGTC	F.
AatI	AGGCCT	AGGCCT	O.
AatII	GACGTC	GACGTC	AFGIKMNORV.
M.AatII	GACGTC	GACGTC	
AauI	TGTACA	TGTACA	
AbaI	TGATCA	TGATCA	
AbeI	CCTCAGC	GCTGAGG	
AbrI	CTCGAG	CTCGAG	
M.AbrI	CTCGAG	CTCGAG	
AcaI	TTCGAA	TTCGAA	
AcaII	GGATCC	GGATCC	
AcaIII	TGCGCA	TGCGCA	
AcaIV	GGCC	GGCC	
AccI	GTMKAC	GTMKAC	ABGJKMNORSU.
M.AccI	GTMKAC	GTMKAC	
AccII	CGCG	CGCG	AJK.
AccIII	TCCGGA	TCCGGA	GJKR.
M.AccIII	TCCGGA	TCCGGA	
Acc16I	TGCGCA	TGCGCA	IV.
Acc36I	ACCTGC	GCAGGT	I.
Acc38I	CCWGG	CCWGG	
Acc65I	GGTACC	GGTACC	FGINRV.
M.Acc65I	GGTACC	GGTACC	
Acc113I	AGTACT	AGTACT	

AccB1I	GGYRCC	GGYRCC	IV.
AccB2I	RGCGCY	RGCGCY	
AccB7I	CCANNNNNTGG	CCANNNNNTGG	IRV.
AccBSI	CCGCTC	GAGCGG	IV.
AccEBI	GGATCC	GGATCC	
AceI	GCWGC	GCWGC	
AceII	GCTAGC	GCTAGC	
AceIII	CAGCTC	GAGCTG	
AciI	CCGC	GCGG	N.
M.AciI	CCGC	CCGC	
AclI	AACGTT	AACGTT	INV.
M.AclI	AACGTT	AACGTT	
AclNI	ACTAGT	ACTAGT	
AclWI	GGATC	GATCC	I.
AcoI	YCCGGR	YCCGGR	I.
AcpI	TTCGAA	TTCGAA	
AcpII	CCANNNNNTGG	CCANNNNNTGG	
AcrI	CYCGRG	CYCGRG	
AcrII	GGTNACC	GGTNACC	
AcsI	RAATTY	RAATTY	IMV.
Acs1371I	GTCGAC	GTCGAC	
Acs1372I	GTCGAC	GTCGAC	
Acs1373I	GTCGAC	GTCGAC	
Acs1421I	GTCGAC	GTCGAC	
Acs1422I	GTCGAC	GTCGAC	
AcuI	CTGAAG	CTTCAG	IN.
M.AcuI	CTGAAG	CTGAAG	
AcuII	CCWGG	CCWGG	
AcvI	CACGTG	CACGTG	QX.
AcyI	GRCGYC	GRCGYC	JM.
AcyII	?	?	
AdeI	CACNNNGTG	CACNNNGTG	F.
AerAI	CTCGAG	CTCGAG	
AeuI	CCWGG	CCWGG	
AfaI	GTAC	GTAC	AK.
Afa22MI	CGATCG	CGATCG	
M.Afa22MI	CGATCG	CGATCG	
Afa16RI	CGATCG	CGATCG	
Afa24RI	GCCGGC	GCCGGC	
AfeI	AGCGCT	AGCGCT	IN.
AfiI	CCNNNNNNNGG	CCNNNNNNNGG	V.
AflI	GGWCC	GGWCC	
AflII	CTTAAG	CTTAAG	AJKNO.
M.AflII	CTTAAG	CTTAAG	
AflIII	ACRYGT	ACRYGT	GMNS.
M.AflIII	ACRYGT	ACRYGT	
AflIV	AGTACT	AGTACT	
Afl83I	TTCGAA	TTCGAA	
Afl83II	GGCC	GGCC	
AgeI	ACCGGT	ACCGGT	GJNR.
M.AgeI	ACCGGT	ACCGGT	
AglI	CCWGG	CCWGG	
AhaI	CCSGG	CCSGG	
AhaII	GRCGYC	GRCGYC	
AhaIII	TTTAAA	TTTAAA	
AhaB1I	GGNCC	GGNCC	
AhaB8I	GGTACC	GGTACC	
AhdI	GACNNNNNGTC	GACNNNNNGTC	GN.
M.AhdI	GACNNNNNGTC	GACNNNNNGTC	
AhlI	ACTAGT	ACTAGT	IV.
AhyI	CCCGGG	CCCGGG	
Ahy45I	?	?	
AhyAI	CTCGAG	CTCGAG	
AimI	?	?	
M.AimAI	?	?	
M.AimAII	?	?	
AinI	CTGCAG	CTGCAG	
AinII	GGATCC	GGATCC	
AitI	AGCGCT	AGCGCT	
AitII	RGATCY	RGATCY	
AitAI	RGATCY	RGATCY	
AjiI	CACGTC	GACGTG	F.
AjnI	CCWGG	CCWGG	I.
AjoI	CTGCAG	CTGCAG	
AjuI	GAANNNNNNNTTGG	CCAANNNNNNNTTC	F.
AjuI	CCAANNNNNNNTTC	GAANNNNNNNTTGG	F.
M.AlaK2I	GATC	GATC	
AleI	CACNNNNGTG	CACNNNNGTG	N.
AlfI	GCANNNNNNTGC	GCANNNNNNTGC	F.
AlfI	GCANNNNNNTGC	GCANNNNNNTGC	F.

AliI	GGATCC	GGATCC	
Ali2882I	CTGCAG	CTGCAG	
Ali12257I	GGATCC	GGATCC	
Ali12258I	GGATCC	GGATCC	
AliAJI	CTGCAG	CTGCAG	
AloI	GAACNNNNNNTCC	GGANNNNNNGTTC	F.
AloI	GGANNNNNNGTTC	GAACNNNNNNTCC	F.
AluI	AGCT	AGCT	ABCFGHIJKMNOQRSUVXY.
M.AluI	AGCT	AGCT	KN.
AlwI	GGATC	GATCC	N.
M.AlwI	GGATC	GGATC	
Alw21I	GWGCWC	GWGCWC	F.
Alw26I	GTCTC	GAGAC	FR.
M.Alw26I	GTCTC	GTCTC	
Alw44I	GTGCAC	GTGCAC	FJMORS.
AlwFI	GAAAYNNNNNRTG	CAYNNNNNRTTTC	
AlwFII	CTCGAG	CTCGAG	
AlwNI	CAGNNNCTG	CAGNNNCTG	N.
AlwXI	GCAGC	GCTGC	
AmaI	TCGCGA	TCGCGA	
I-AmaI	?	?	
Ama87I	CYCGRG	CYCGRG	IV.
AmeI	GTGCAC	GTGCAC	
AmeII	GCCGGC	GCCGGC	
AniI	?	?	
I-AniI	TTGAGGAGGTTTCTCTGTAAATAA	TTATTTACAGAGAAACCTCCTCAA	
AniAI	?	?	
AniMI	GCCGGC	GCCGGC	
AocI	CCTNAGG	CCTNAGG	
AocII	GDGCHC	GDGCHC	
AorI	CCWGG	CCWGG	
Aor13HI	TCCGGA	TCCGGA	K.
Aor51HI	AGCGCT	AGCGCT	AK.
AosI	TGCGCA	TGCGCA	
AosII	GRCGYC	GRCGYC	
AosIII	CCGCGG	CCGCGG	
ApaI	GGGCCC	GGGCCC	ABFGIJKMNOQRSUVX.
M.ApaI	GGGCCC	GGGCCC	
ApaBI	GCANNNNNTGC	GCANNNNNTGC	
ApaCI	GGATCC	GGATCC	
ApaDI	?	?	
ApaLI	GTGCAC	GTGCAC	AKNU.
M.ApaLI	GTGCAC	GTGCAC	
ApaORI	CCWGG	CCWGG	
Apc202I	?	?	
ApcTR183I	TGCGCA	TGCGCA	
ApeI	ACGCGT	ACGCGT	
ApeAI	GCCGGC	GCCGGC	
ApeKI	GCWGC	GCWGC	N.
I-ApeKI	GCAAGGCTGAAACTTAAAGG	CCTTTAAGTTTCAGCCTTGC	
M.ApeKI	GCWGC	GCWGC	
ApiI	CTGCAG	CTGCAG	
ApoI	RAATTY	RAATTY	N.
M.ApoI	RAATTY	RAATTY	
AprI	GCCGGC	GCCGGC	
ApuI	GGNCC	GGNCC	
Apu16I	ATCGAT	ATCGAT	
ApyI	CCWGG	CCWGG	
AquI	CYCGRG	CYCGRG	
M.AquI	CYCGRG	CYCGRG	
AscI	GGCGCGCC	GGCGCGCC	GN.
M.AscI	GGCGCGCC	GGCGCGCC	
AseI	ATTAAT	ATTAAT	JNO.
M.AseI	ATTAAT	ATTAAT	
AseII	CCSGG	CCSGG	
M.AseII	CCSGG	CCSGG	
AsiI	GGATCC	GGATCC	
AsiAI	ACCGGT	ACCGGT	
AsiGI	ACCGGT	ACCGGT	IV.
AsiSI	GCGATCGC	GCGATCGC	N.
M.AsiSI	GCGATCGC	GCGATCGC	
AsnI	ATTAAT	ATTAAT	
AspI	GACNNNGTC	GACNNNGTC	M.
Asp1I	CCSGG	CCSGG	
Asp10I	?	?	
Asp14I	ATCGAT	ATCGAT	
Asp15I	CTCGAG	CTCGAG	
Asp17I	RGATCY	RGATCY	
Asp22I	RGATCY	RGATCY	
Asp28I	?	?	

Asp36I	CTGCAG	CTGCAG	
Asp37I	ATCGAT	ATCGAT	
Asp47I	CTCGAG	CTCGAG	
Asp52I	AAGCTT	AAGCTT	
Asp54I	?	?	
Asp78I	AGGCCT	AGGCCT	
Asp86I	ATCGAT	ATCGAT	
Asp86II	?	?	
Asp90I	ACRYGT	ACRYGT	
Asp90II	?	?	
Asp123I	ATCGAT	ATCGAT	
Asp123II	?	?	
Asp130I	ATCGAT	ATCGAT	
Asp697I	GGWCC	GGWCC	
Asp700I	GAANNNNTTC	GAANNNNTTC	M.
Asp703I	CTCGAG	CTCGAG	
Asp707I	ATCGAT	ATCGAT	
Asp708I	CTGCAG	CTGCAG	
Asp713I	CTGCAG	CTGCAG	
Asp718I	GGTACC	GGTACC	M.
Asp742I	GGCC	GGCC	
Asp745I	GGWCC	GGWCC	
Asp748I	CCGG	CCGG	
Asp763I	AGTACT	AGTACT	
Asp3065I	AAGCTT	AAGCTT	
AspAI	GGTNACC	GGTNACC	
AspA2I	CCTAGG	CCTAGG	IV.
Asp202A1I	?	?	
Asp202A135I	?	?	
AspBI	CYCGRG	CYCGRG	
AspBII	GGWCC	GGWCC	
AspCNI	GCCGC	GCGGC	
M.AspCNI	GCSGC	GCSGC	
AspDI	CYCGRG	CYCGRG	
AspDII	GGWCC	GGWCC	
AspEI	GACNNNNNGTC	GACNNNNNGTC	M.
AspHI	GWGCWC	GWGCWC	
Asp1HI	RGATCY	RGATCY	
Asp2HI	CCWGG	CCWGG	
Asp5HI	GCATGC	GCATGC	
Asp6HI	RGATCY	RGATCY	
Asp8HI	RGATCY	RGATCY	
Asp10HI	TTCGAA	TTCGAA	
Asp10HII	CCANNNNNTGG	CCANNNNNTGG	
Asp14HI	RGATCY	RGATCY	
Asp16HI	GTAC	GTAC	
Asp17HI	GTAC	GTAC	
Asp18HI	GTAC	GTAC	
Asp21HI	RGATCY	RGATCY	
Asp26HI	GAATGC	GCATTC	
Asp27HI	GAATGC	GCATTC	
Asp29HI	GTAC	GTAC	
Asp32HI	CCGCGG	CCGCGG	
Asp35HI	GAATGC	GCATTC	
Asp36HI	GAATGC	GCATTC	
Asp40HI	GAATGC	GCATTC	
Asp50HI	GAATGC	GCATTC	
AspJI	GACGTC	GACGTC	
AspLEI	GCGC	GCGC	IV.
AspMI	AGGCCT	AGGCCT	
AspMDI	GATC	GATC	
AspNI	GGNNCC	GGNNCC	
AspS9I	GGNCC	GGNCC	IV.
AspTI	CTGCAG	CTGCAG	
AspTII	GGATCC	GGATCC	
AspTIII	GGCC	GGCC	
AssI	AGTACT	AGTACT	U.
AstWI	GRCGYC	GRCGYC	
AsuI	GGNCC	GGNCC	
AsuII	TTCGAA	TTCGAA	C.
AsuIII	GRCGYC	GRCGYC	
AsuC2I	CCSGG	CCSGG	I.
AsuHPI	GGTGA	TCACC	IV.
AsuMBI	GATC	GATC	
AsuNHI	GCTAGC	GCTAGC	IV.
AsuSAI	CCTNAGG	CCTNAGG	
AteI	CCATGG	CCATGG	
M.AthIII	?	?	
M.AthDRM2	?	?	
M.AthDnmt1A	?	?	

M.AthDnmt1B	?	?	
M.AthVIII	?	?	
AtsI	GACNNNGTC	GACNNNGTC	
AtuII	CCWGG	CCWGG	
Atu1I	CCWGG	CCWGG	
Atu1II	GGATCC	GGATCC	
AtuAI	?	?	
AtuBI	CCWGG	CCWGG	
AtuBVI	?	?	
M.AtuCI	GANTC	GANTC	
AtuIAMI	?	?	
AtuSI	TGATCA	TGATCA	
AvaI	CYCGRG	CYCGRG	ABGJKMNORSUX.
M.AvaI	CYCGRG	CYCGRG	
AvaII	GGWCC	GGWCC	AGJKMNRSY.
M.AvaII	GGWCC	GGWCC	
AvaIII	ATGCAT	ATGCAT	
M.AvaIII	ATGCAT	ATGCAT	
M.AvaV	GATC	GATC	
M.AvaVI	GATC	GATC	
M.AvaVII	GGCC	GGCC	
M.AvaVIII	CGATCG	CGATCG	
M.AvaIX	RCCGGY	RCCGGY	
Ava458I	YGGCCR	YGGCCR	
AvaBORF3498	?	?	
M.AvaBORF3498	?	?	
AvcI	GGNCC	GGNCC	
AviI	TTCGAA	TTCGAA	
AviII	TGCGCA	TGCGCA	M.
AvoI	RCATGY	RCATGY	
AvrI	CYCGRG	CYCGRG	
M.AvrI	CYCGRG	CYCGRG	
AvrII	CCTAGG	CCTAGG	N.
M.AvrII	CCTAGG	CCTAGG	
AvrBI	GGCC	GGCC	
AvrBII	CCTAGG	CCTAGG	
AxyI	CCTNAGG	CCTNAGG	J.
M.BabI	GANTC	GANTC	
BacI	CCGCGG	CCGCGG	
Bac36I	GGNCC	GGNCC	
Bac465I	CCGCGG	CCGCGG	
BadI	CTCGAG	CTCGAG	
BaeI	ACNNNNGTAYC	GRTACNNNNGT	N.
BaeI	GRTACNNNNGT	ACNNNNGTAYC	N.
M.BaeI	ACNNNNGTAYC	ACNNNNGTAYC	
BalI	TGGCCA	TGGCCA	AJKR.
M.BalI	TGGCCA	TGGCCA	
Bal228I	GGNCC	GGNCC	
Bal475I	GGCC	GGCC	
Bal3006I	GGCC	GGCC	
BamFI	GGATCC	GGATCC	
BamGI	CAGCTG	CAGCTG	
BamHI	GGATCC	GGATCC	ABCFGHIJKMNOQRSUVXY.
M.BamHI	GGATCC	GGATCC	KN.
M.BamHII	GGATCC	GGATCC	
BamKI	GGATCC	GGATCC	
BamNI	GGATCC	GGATCC	
BamNxI	GGWCC	GGWCC	
BanI	GGYRCC	GGYRCC	NORU.
M.BanI	GGYRCC	GGYRCC	
BanII	GRGCYC	GRGCYC	AGKMNOQRSX.
M.BanII	GRGCYC	GRGCYC	
BanIII	ATCGAT	ATCGAT	O.
M.BanIII	ATCGAT	ATCGAT	
BanAI	GGCC	GGCC	
BasI	CCANNNNNTGG	CCANNNNNTGG	
I-BasI	AGTAATGAGCCTAACGCTCAGCAA	TTGCTGAGCGTTAGGCTCATTACT	
BauI	CACGAG	CTCGTG	F.
BavI	CAGCTG	CAGCTG	
BavAI	CAGCTG	CAGCTG	
BavAII	GGNCC	GGNCC	
BavBI	CAGCTG	CAGCTG	
BavBII	GGNCC	GGNCC	
BavCI	ATCGAT	ATCGAT	
BazI	ATCGAT	ATCGAT	
Bba179I	WCCGGW	WCCGGW	
BbeI	GGCGCC	GGCGCC	AK.
BbeII	?	?	
BbeAI	GGCGCC	GGCGCC	
BbeAII	?	?	

BbeSI	?	?	
BbfI	CTCGAG	CTCGAG	
Bbf7411I	TCCGGA	TCCGGA	
BbiI	CTGCAG	CTGCAG	
BbiII	GRCGYC	GRCGYC	
BbiIII	CTCGAG	CTCGAG	
BbiIV	?	?	
Bbi24I	ACGCGT	ACGCGT	
BboI	?	?	
BbrI	AAGCTT	AAGCTT	
Bbr7I	GAAGAC	GTCTTC	
BbrAI	AAGCTT	AAGCTT	
BbrPI	CACGTG	CACGTG	MO.
BbsI	GAAGAC	GTCTTC	N.
BbtI	GCGC	GCGC	
BbuI	GCATGC	GCATGC	R.
M.Bbu297I	CCWGG	CCWGG	
BbvI	GCAGC	GCTGC	N.
M.BbvI	GCAGC	GCAGC	
BbvII	GAAGAC	GTCTTC	
Bbv12I	GWGCWC	GWGCWC	IV.
Bbv16II	GAAGAC	GTCTTC	
BbvAI	GAANNNNTTC	GAANNNNTTC	
BbvAII	ATCGAT	ATCGAT	
BbvAIII	TCCGGA	TCCGGA	
BbvBI	GGYRCC	GGYRCC	
BbvCI	CCTCAGC	GCTGAGG	N.
M1.BbvCI	CCTCAGC	CCTCAGC	
M2.BbvCI	CCTCAGC	CCTCAGC	
M.BbvSI	GCWGC	GCWGC	
BcaI	GCGC	GCGC	
Bca77I	WCCGGW	WCCGGW	
Bca1259I	GGATCC	GGATCC	
BccI	CCATC	GATGG	N.
M1.BccI	CCATC	CCATC	
M2.BccI	CCATC	CCATC	
Bce4I	GCNNNNNNNGC	GCNNNNNNNGC	
Bce22I	GGNCC	GGNCC	
Bce71I	GGCC	GGCC	
Bce83I	CTTGAG	CTCAAG	
Bce170I	CTGCAG	CTGCAG	
Bce243I	GATC	GATC	
Bce751I	GGATCC	GGATCC	
Bce1229I	?	?	
Bce1247I	GCNNNNNNNGC	GCNNNNNNNGC	
M.Bce1247I	GCNNNNNNNGC	GCNNNNNNNGC	
Bce14579I	?	?	
Bce31293I	CGCG	CGCG	
BceAI	ACGGC	GCCGT	N.
M1.BceAI	ACGGC	ACGGC	
M2.BceAI	ACGGC	ACGGC	
BceBI	CGCG	CGCG	
BceCI	GCNNNNNNNGC	GCNNNNNNNGC	
BceDI	TGATCA	TGATCA	
BceRI	CGCG	CGCG	
BceSI	?	?	
M.BceSI	?	?	
BcefI	ACGGC	GCCGT	
BcgI	CGANNNNNNTGC	GCANNNNNNTCG	N.
BcgI	GCANNNNNNTCG	CGANNNNNNTGC	N.
BchI	GCAGC	GCTGC	
M.BchI	GCAGC	GCAGC	
Bci29I	ATCGAT	ATCGAT	
BciAI	?	?	
BciBI	ATCGAT	ATCGAT	
BciBII	CCWGG	CCWGG	
BciVI	GTATCC	GGATAC	N.
BclI	TGATCA	TGATCA	CFGJMNORSUY.
M.BclI	TGATCA	TGATCA	
BcmI	ATCGAT	ATCGAT	
BcnI	CCSGG	CCSGG	FK.
M1.BcnI	CCSGG	CCSGG	
M2.BcnI	CCSGG	CCSGG	
BcoI	CYCGRG	CYCGRG	
Bco5I	CTCTTC	GAAGAG	
Bco6I	TGCGCA	TGCGCA	
Bco27I	CCGG	CCGG	
Bco33I	GGCC	GGCC	
Bco35I	CTGGAG	CTCCAG	
Bco63I	GATNNNNATC	GATNNNNATC	

Bco79I	ATCGAT	ATCGAT	
Bco102I	TGATCA	TGATCA	
Bco102II	GAAGAC	GTCTTC	
Bco116I	CTCTTC	GAAGAG	
Bco118I	RCCGGY	RCCGGY	
Bco163I	CTRYAG	CTRYAG	
Bco631I	GATNNNNATC	GATNNNNATC	
Bco10278I	GGATCC	GGATCC	
BcoAI	CACGTG	CACGTG	
BcoKI	CTCTTC	GAAGAG	
M1.BcoKI	CTCTTC	CTCTTC	
M2.BcoKI	CTCTTC	CTCTTC	
BcoSI	CTCTTC	GAAGAG	
BcrI	GGNNCC	GGNNCC	
BcrAI	CTCTTC	GAAGAG	
BctI	ACGGC	GCCGT	
BcuI	ACTAGT	ACTAGT	F.
BcuAI	GGWCC	GGWCC	
BdaI	TGANNNNNNTCA	TGANNNNNNTCA	F.
BdaI	TGANNNNNNTCA	TGANNNNNNTCA	F.
BdiI	ATCGAT	ATCGAT	
M.BdiI	ATCGAT	ATCGAT	
BdiSI	CTRYAG	CTRYAG	
BecAI	?	?	
BecAII	GGCC	GGCC	
BepI	CGCG	CGCG	
M.BepI	CGCG	CGCG	
BetI	WCCGGW	WCCGGW	
BfaI	CTAG	CTAG	N.
BfiI	ACTGGG	CCCAGT	F.
M1.BfiI	ACTGGG	ACTGGG	
M2.BfiI	ACTGGG	ACTGGG	
Bfi57I	GATC	GATC	
Bfi89I	YGGCCR	YGGCCR	
Bfi105I	GGNCC	GGNCC	
Bfi458I	GGCC	GGCC	
Bfi2411I	?	?	
BfiSHI	GATC	GATC	
BflI	CCNNNNNNNGG	CCNNNNNNNGG	
M.BflBF4I	GCSGC	GCSGC	
BfmI	CTRYAG	CTRYAG	F.
BfrI	CTTAAG	CTTAAG	MO.
BfrAI	ATCGAT	ATCGAT	
BfrBI	ATGCAT	ATGCAT	
BfrCI	ATGCAT	ATGCAT	
BfuI	GTATCC	GGATAC	F.
Bfu1570I	GWGCWC	GWGCWC	
BfuAI	ACCTGC	GCAGGT	N.
M1.BfuAI	ACCTGC	ACCTGC	
M2.BfuAI	ACCTGC	ACCTGC	
BfuCI	GATC	GATC	N.
BgiI	GACNNNGTC	GACNNNGTC	
BglI	GCCNNNNNGGC	GCCNNNNNGGC	ACFGHIJKMNOQRSUVXY.
M.BglI	GCCNNNNNGGC	GCCNNNNNGGC	
BglII	AGATCT	AGATCT	ABCFGHIJKMNOQRSUVXY.
M.BglII	AGATCT	AGATCT	
BhaI	GCATC	GATGC	
M1.BhaI	GCATC	GCATC	
M2.BhaI	GCATC	GCATC	
BhaII	GGCC	GGCC	
M.BhaII	GGCC	GGCC	
BheI	GCCGGC	GCCGGC	
BimI	TTCGAA	TTCGAA	
Bim19I	TTCGAA	TTCGAA	
Bim19II	GGCC	GGCC	
BinI	GGATC	GATCC	
BinSI	CCWGG	CCWGG	
BinSII	GGCGCC	GGCGCC	
BisI	GCNGC	GCNGC	I.
Bka1125I	GDGCHC	GDGCHC	
Bla7920I	TCCGGA	TCCGGA	
BlfI	TCCGGA	TCCGGA	U.
BliI	GGCC	GGCC	
Bli41I	ATCGAT	ATCGAT	
Bli49I	GGTCTC	GAGACC	
Bli86I	ATCGAT	ATCGAT	
Bli161I	GGTCTC	GAGACC	
Bli576I	ATCGAT	ATCGAT	
Bli576II	GGTCTC	GAGACC	
Bli585I	ATCGAT	ATCGAT	

Bli643I	CCTNAGG	CCTNAGG	
Bli736I	GGTCTC	GAGACC	
M.Bli736I	GGTCTC	GGTCTC	
Bli5508I	GGTCTC	GAGACC	
Bli11054I	?	?	
BliAI	ATCGAT	ATCGAT	
BliHKI	CCTNAGG	CCTNAGG	
BliRI	ATCGAT	ATCGAT	
BlnI	CCTAGG	CCTAGG	AKMS.
BloI	?	?	
BloHI	RGATCY	RGATCY	
BloHII	CTGCAG	CTGCAG	
BloHIII	CTGCAG	CTGCAG	
BlpI	GCTNAGC	GCTNAGC	N.
M.BlpI	GCTNAGC	GCTNAGC	
BluI	CTCGAG	CTCGAG	
BluII	GGCC	GGCC	
BmaI	CGATCG	CGATCG	
M.BmaI	CGATCG	CGATCG	
BmaAI	CGATCG	CGATCG	
BmaBI	CGATCG	CGATCG	
BmaCI	CGATCG	CGATCG	
BmaDI	CGATCG	CGATCG	
BmaHI	GAATGC	GCATTC	
M.BmaPhiE125I	?	?	
M.BmaPhiE125I	?	?	
BmcAI	AGTACT	AGTACT	V.
BmeI	?	?	
Bme05I	GGYRCC	GGYRCC	
Bme12I	GATC	GATC	
Bme18I	GGWCC	GGWCC	IV.
Bme46I	GGCC	GGCC	
Bme74I	GGCC	GGCC	
Bme142I	RGCGCY	RGCGCY	
Bme205I	?	?	
Bme216I	GGWCC	GGWCC	
M.Bme216I	GGWCC	GGWCC	
Bme361I	GGCC	GGCC	
Bme585I	CCCGC	GCGGG	
Bme899I	?	?	
Bme1390I	CCNGG	CCNGG	F.
Bme1580I	GKGCMC	GKGCMC	N.
Bme2095I	CCWGG	CCWGG	
Bme2494I	GATC	GATC	
BmeBI	CTGCAG	CTGCAG	
BmeRI	GACNNNNNGTC	GACNNNNNGTC	V.
BmeTI	TGATCA	TGATCA	
M.BmeTI	TGATCA	TGATCA	
BmeT110I	CYCGRG	CYCGRG	K.
BmeU1594I	GGCC	GGCC	
BmgI	GKGCCC	GGGCMC	
BmgAI	GKGCMC	GKGCMC	
BmgBI	CACGTC	GACGTG	N.
BmgT120I	GGNCC	GGNCC	K.
BmiI	GGNNCC	GGNNCC	V.
I-BmoI	GAGTAAGAGCCCGTAGTAATGACATGGC	GCCATGTCATTACTACGGGCTCTTACTC	
BmpI	GGWCC	GGWCC	
BmrI	ACTGGG	CCCAGT	N.
M1.BmrI	ACTGGG	ACTGGG	
M2.BmrI	ACTGGG	ACTGGG	
BmrFI	CCNGG	CCNGG	V.
BmtI	GCTAGC	GCTAGC	INV.
BmuI	ACTGGG	CCCAGT	I.
BmyI	GDGCHC	GDGCHC	
BnaI	GGATCC	GGATCC	
M.BnaI	GGATCC	GGATCC	
BoxI	GACNNNNGTC	GACNNNNGTC	F.
BpaI	?	?	
Bpa34I	AGTACT	AGTACT	
Bpa36I	GGCC	GGCC	
Bpa36II	CTNAG	CTNAG	
BpcI	CTRYAG	CTRYAG	U.
BpeI	AAGCTT	AAGCTT	
BpiI	GAAGAC	GTCTTC	F.
BplI	GAGNNNNNCTC	GAGNNNNNCTC	F.
BplI	GAGNNNNNCTC	GAGNNNNNCTC	F.
BpmI	CTGGAG	CTCCAG	IN.
M.BpmI	CTGGAG	CTGGAG	
BpnI	?	?	
BpoAI	ATTAAT	ATTAAT	

BprI	?	?	
BpsI	GGNCC	GGNCC	
BptI	CCWGG	CCWGG	U.
BpuI	GRGCYC	GRGCYC	
Bpu10I	CCTNAGC	GCTNAGG	FINV.
M1.Bpu10I	CCTNAGC	CCTNAGC	
M2.Bpu10I	CCTNAGC	CCTNAGC	
Bpu14I	TTCGAA	TTCGAA	IV.
Bpu86I	GCCNNNNNGGC	GCCNNNNNGGC	
Bpu95I	CGCG	CGCG	
Bpu1102I	GCTNAGC	GCTNAGC	AFK.
Bpu1268I	CCTNNNNNAGG	CCTNNNNNAGG	
Bpu1811I	GCNGC	GCNGC	
Bpu1831I	TACGTA	TACGTA	
BpuAI	GAAGAC	GTCTTC	M.
BpuAmI	GAGCTC	GAGCTC	
BpuB5I	CGTACG	CGTACG	
BpuCI	GGCGGA	TCCGCC	
BpuDI	CCTNAGC	GCTNAGG	
BpuEI	CTTGAG	CTCAAG	N.
BpuFI	GGATC	GATCC	
BpuGI	RGATCY	RGATCY	
BpuGCI	GCTNAGC	GCTNAGC	
BpuHI	TTCGAA	TTCGAA	
BpuJI	CCCGT	ACGGG	
BpuMI	CCSGG	CCSGG	V.
BpuNI	GGGAC	GTCCC	
BpuSI	GGGAC	GTCCC	
M1.BpuSI	GGGAC	GGGAC	
M2.BpuSI	GGGAC	GGGAC	
BpvUI	CGATCG	CGATCG	V.
BsaI	GGTCTC	GAGACC	N.
M1.BsaI	GGTCTC	GGTCTC	
M2.BsaI	GGTCTC	GGTCTC	
Bsa29I	ATCGAT	ATCGAT	I.
BsaAI	YACGTR	YACGTR	N.
M.BsaAI	YACGTR	YACGTR	
BsaBI	GATNNNNATC	GATNNNNATC	N.
BsaCI	CCNGG	CCNGG	
BsaDI	GGATCC	GGATCC	
BsaEI	GGNNCC	GGNNCC	
BsaFI	CTTAAG	CTTAAG	
BsaGI	GWGCWC	GWGCWC	
BsaHI	GRCGYC	GRCGYC	N.
BsaJI	CCNNGG	CCNNGG	N.
M.BsaJI	CCNNGG	CCNNGG	
BsaKI	GTTAAC	GTTAAC	
BsaLI	AGCT	AGCT	
BsaMI	GAATGC	GCATTC	GR.
BsaNI	CCWGG	CCWGG	
BsaNII	CTGCAG	CTGCAG	
BsaOI	CGRYCG	CGRYCG	
BsaPI	GATC	GATC	
BsaQI	CTGCAG	CTGCAG	
BsaRI	GGCC	GGCC	
BsaRII	?	?	
BsaSI	GGNCC	GGNCC	
BsaTI	TGCGCA	TGCGCA	
BsaUI	GCAGC	GCTGC	
BsaVI	GAAGAC	GTCTTC	
BsaWI	WCCGGW	WCCGGW	N.
M.BsaWI	WCCGGW	WCCGGW	
BsaXI	ACNNNNNCTCC	GGAGNNNNNGT	N.
BsaXI	GGAGNNNNNGT	ACNNNNNCTCC	N.
BsaZI	CCGG	CCGG	
BsbI	CAACAC	GTGTTG	
BscI	ATCGAT	ATCGAT	
Bsc4I	CCNNNNNNNGG	CCNNNNNNNGG	I.
Bsc91I	GAAGAC	GTCTTC	
Bsc107I	CCNNNNNNNGG	CCNNNNNNNGG	
Bsc217I	GATATC	GATATC	
BscAI	GCATC	GATGC	
BscBI	GGNNCC	GGNNCC	
BscCI	GAATGC	GCATTC	
BscDI	CTGCAG	CTGCAG	
BscEI	GCGCGC	GCGCGC	
BscFI	GATC	GATC	
BscGI	CCCGT	ACGGG	
M1.BscGI	CCCGT	CCCGT	
M2.BscGI	CCCGT	CCCGT	

BscHI	ACTGG	CCAGT	
BscJI	CCANNNNNNTGG	CCANNNNNNTGG	
BscKI	GAAGAC	GTCTTC	
BscLI	CTTAAG	CTTAAG	
BscMI	GRGCYC	GRGCYC	
BscNI	CGRYCG	CGRYCG	
BscOI	GCATGC	GCATGC	
BscPI	CTNAG	CTNAG	
BscQI	GGCC	GGCC	
BscQII	GTCTC	GAGAC	
BscRI	RCCGGY	RCCGGY	
BscSI	RGATCY	RGATCY	
BscTI	CCGCGG	CCGCGG	
BscUI	GCATC	GATGC	
BscVI	ATCGAT	ATCGAT	
BscWI	GGGAC	GTCCC	
BseI	GGCC	GGCC	
BseII	GTTAAC	GTTAAC	
Bse1I	ACTGG	CCAGT	IV.
Bse8I	GATNNNNATC	GATNNNNATC	IV.
Bse9I	GGCC	GGCC	
Bse15I	CYCGRG	CYCGRG	
Bse16I	CCWGG	CCWGG	
Bse17I	CCWGG	CCWGG	
Bse19I	CCATGG	CCATGG	
Bse21I	CCTNAGG	CCTNAGG	IV.
Bse23I	CCNNNNNNNGG	CCNNNNNNNGG	
Bse24I	CCWGG	CCWGG	
Bse54I	GGNCC	GGNCC	
Bse59I	GGTNACC	GGTNACC	
Bse64I	GGTNACC	GGTNACC	
Bse118I	RCCGGY	RCCGGY	IV.
Bse126I	GGCC	GGCC	
Bse631I	GATNNNNATC	GATNNNNATC	
Bse634I	RCCGGY	RCCGGY	
M.Bse634I	RCCGGY	RCCGGY	
BseAI	TCCGGA	TCCGGA	CM.
BseBI	CCWGG	CCWGG	C.
BseB631I	GCCNNNNNGGC	GCCNNNNNGGC	
BseB631II	AGATCT	AGATCT	
BseCI	ATCGAT	ATCGAT	C.
M.BseCI	ATCGAT	ATCGAT	
BseDI	CCNNGG	CCNNGG	F.
M.BseDI	CCNNGG	CCNNGG	
Bse3DI	GCAATG	CATTGC	IV.
BseEI	?	?	
BseFI	?	?	
BseGI	GGATG	CATCC	F.
BseG73I	CCTNAGG	CCTNAGG	
BseHI	AAGCTT	AAGCTT	
BseJI	GATNNNNATC	GATNNNNATC	F.
BseKI	GCAGC	GCTGC	
BseLI	CCNNNNNNNGG	CCNNNNNNNGG	F.
BseMI	GCAATG	CATTGC	F.
BseMII	CTCAG	CTGAG	F.
M.BseMII	?	?	
BseNI	ACTGG	CCAGT	FG.
BsePI	GCGCGC	GCGCGC	IV.
BseQI	GGCC	GGCC	
BseRI	GAGGAG	CTCCTC	N.
M.BseRI	GAGGAG	GAGGAG	
BseSI	GKGCMC	GKGCMC	F.
BseTI	?	?	
BseT9I	GGTNACC	GGTNACC	
BseT10I	GGTNACC	GGTNACC	
BseWI	?	?	
BseXI	GCAGC	GCTGC	F.
BseX3I	CGGCCG	CGGCCG	IV.
BseYI	CCCAGC	GCTGGG	N.
M.BseYI	CCCAGC	CCCAGC	
BseZI	CTCTTC	GAAGAG	
BsgI	GTGCAG	CTGCAC	N.
M.BsgI	GTGCAG	GTGCAG	
BshI	GGCC	GGCC	
Bsh45I	GWGCWC	GWGCWC	
Bsh1236I	CGCG	CGCG	F.
Bsh1285I	CGRYCG	CGRYCG	F.
Bsh1365I	GATNNNNATC	GATNNNNATC	
BshAI	GGCC	GGCC	
Bsh108AI	ATCGAT	ATCGAT	

BshBI	GGCC	GGCC	
BshCI	GGCC	GGCC	
BshDI	GGCC	GGCC	
BshEI	GGCC	GGCC	
BshFI	GGCC	GGCC	C.
BshGI	CCWGG	CCWGG	
BshHI	AGTACT	AGTACT	
BshKI	GGNCC	GGNCC	
BshLI	GATATC	GATATC	
BshMI	CCGG	CCGG	
BshNI	GGYRCC	GGYRCC	F.
BshTI	ACCGGT	ACCGGT	F.
BshVI	ATCGAT	ATCGAT	V.
BsiI	CACGAG	CTCGTG	
BsiAI	GGCC	GGCC	
BsiBI	GATNNNNATC	GATNNNNATC	
BsiCI	TTCGAA	TTCGAA	
BsiDI	GGCC	GGCC	
BsiEI	CGRYCG	CGRYCG	N.
BsiFI	?	?	
BsiGI	TCCGGA	TCCGGA	
BsiHI	GGCC	GGCC	
BsiHKAI	GWGCWC	GWGCWC	N.
BsiHKCI	CYCGRG	CYCGRG	QX.
BsiJI	?	?	
BsiKI	GGTNACC	GGTNACC	
BsiLI	CCWGG	CCWGG	
BsiMI	TCCGGA	TCCGGA	
BsiNI	?	?	
BsiOI	TCCGGA	TCCGGA	
BsiPI	?	?	
BsiQI	TGATCA	TGATCA	
BsiRI	?	?	
BsiSI	CCGG	CCGG	C.
BsiTI	?	?	
BsiUI	CCWGG	CCWGG	
BsiVI	CCWGG	CCWGG	
BsiWI	CGTACG	CGTACG	MNO.
M.BsiWI	CGTACG	CGTACG	
BsiXI	ATCGAT	ATCGAT	
BsiYI	CCNNNNNNNGG	CCNNNNNNNGG	M.
BsiZI	GGNCC	GGNCC	
BslI	CCNNNNNNNGG	CCNNNNNNNGG	GN.
M.BslI	CCNNNNNNNGG	CCNNNNNNNGG	
BslFI	GGGAC	GTCCC	I.
BsmI	GAATGC	GCATTC	JMNOS.
M1.BsmI	GAATGC	GAATGC	
M2.BsmI	GAATGC	GAATGC	
Bsm6I	GWGCWC	GWGCWC	
BsmAI	GTCTC	GAGAC	N.
M.BsmAI	GTCTC	GTCTC	
BsmBI	CGTCTC	GAGACG	N.
M.BsmBI	CGTCTC	CGTCTC	
BsmCI	ACNNNNNCTCC	GGAGNNNNNGT	
BsmDI	ACNNNNNCTCC	GGAGNNNNNGT	
BsmEI	GAGTC	GACTC	
BsmFI	GGGAC	GTCCC	N.
M1.BsmFI	GGGAC	GGGAC	
M2.BsmFI	GGGAC	GGGAC	
BsmGI	TGTACA	TGTACA	
BsmGII	AAGCTT	AAGCTT	
BsmHI	RGCGCY	RGCGCY	
BsmNI	GCATC	GATGC	
BsmPI	GWGCWC	GWGCWC	
BsmRI	TGTACA	TGTACA	
BsmSI	CCWWGG	CCWWGG	
BsmWI	CGTACG	CGTACG	
BsmXI	ACNNNNNCTCC	GGAGNNNNNGT	
BsmXII	GATC	GATC	
BsmYI	CCNNNNNNNGG	CCNNNNNNNGG	
BsnI	GGCC	GGCC	V.
BsoI	CCNGG	CCNGG	
Bso31I	GGTCTC	GAGACC	IV.
BsoAI	GATATC	GATATC	
BsoBI	CYCGRG	CYCGRG	N.
M.BsoBI	CYCGRG	CYCGRG	
BsoCI	GDGCHC	GDGCHC	
BsoDI	CGGCCG	CGGCCG	
BsoEI	CCTNNNNNAGG	CCTNNNNNAGG	
BsoFI	GCNGC	GCNGC	

BsoGI	CCWGG	CCWGG	
BsoGII	?	?	
BsoHI	ACTGG	CCAGT	
BsoJI	GCCNNNNNGGC	GCCNNNNNGGC	
BsoKI	CCNNGG	CCNNGG	
BsoMAI	GTCTC	GAGAC	
BsoPI	GCGCGC	GCGCGC	
BsoSI	AGTACT	AGTACT	
BspI	GATC	GATC	
M.BspI	GATC	GATC	
Bsp2I	ATCGAT	ATCGAT	
Bsp4I	ATCGAT	ATCGAT	
Bsp5I	CCGG	CCGG	
Bsp6I	GCNGC	GCNGC	
M.Bsp6I	GCNGC	GCNGC	
Bsp6II	CTGAAG	CTTCAG	
Bsp7I	CCSGG	CCSGG	
Bsp8I	CCSGG	CCSGG	
Bsp9I	GATC	GATC	
Bsp12I	CCGCGG	CCGCGG	
Bsp12II	?	?	
Bsp13I	TCCGGA	TCCGGA	IV.
Bsp16I	GATATC	GATATC	
Bsp17I	CTGCAG	CTGCAG	
Bsp18I	GATC	GATC	
Bsp19I	CCATGG	CCATGG	IV.
Bsp21I	RCCGGY	RCCGGY	
Bsp22I	CTGGAG	CTCCAG	
Bsp23I	GGCC	GGCC	
Bsp24I	GACNNNNNNTGG	CCANNNNNNGTC	
Bsp24I	CCANNNNNNGTC	GACNNNNNNTGG	
Bsp28I	CTGGAG	CTCCAG	
Bsp29I	GGNNCC	GGNNCC	
Bsp30I	GGATCC	GGATCC	
Bsp42I	?	?	
Bsp43I	CTGCAG	CTGCAG	
Bsp44I	CCWGG	CCWGG	
Bsp44II	GGCC	GGCC	
Bsp46I	GGATCC	GGATCC	
Bsp47I	CCGG	CCGG	
Bsp48I	CCGG	CCGG	
Bsp49I	GATC	GATC	
Bsp50I	CGCG	CGCG	
M.Bsp50I	CGCG	CGCG	
Bsp51I	GATC	GATC	
Bsp52I	GATC	GATC	
Bsp53I	CCNGG	CCNGG	
Bsp54I	GATC	GATC	
Bsp55I	CCSGG	CCSGG	
Bsp56I	CCWGG	CCWGG	
Bsp57I	GATC	GATC	
Bsp58I	GATC	GATC	
Bsp59I	GATC	GATC	
Bsp60I	GATC	GATC	
Bsp61I	GATC	GATC	
Bsp63I	CTGCAG	CTGCAG	
Bsp64I	GATC	GATC	
Bsp65I	GATC	GATC	
Bsp66I	GATC	GATC	
Bsp67I	GATC	GATC	
Bsp68I	TCGCGA	TCGCGA	F.
Bsp70I	CGCG	CGCG	
Bsp71I	GGWCC	GGWCC	
Bsp72I	GATC	GATC	
Bsp73I	CCNGG	CCNGG	
Bsp74I	GATC	GATC	
Bsp76I	GATC	GATC	
Bsp78I	CTGCAG	CTGCAG	
Bsp81I	CTGCAG	CTGCAG	
Bsp82I	TTCGAA	TTCGAA	
Bsp84I	ATCGAT	ATCGAT	
Bsp87I	CACGTG	CACGTG	
Bsp90I	TTCGAA	TTCGAA	
Bsp90II	GGATCC	GGATCC	
Bsp91I	GATC	GATC	
Bsp92I	CTCGAG	CTCGAG	
Bsp93I	CTGCAG	CTGCAG	
Bsp98I	GGATCC	GGATCC	
M.Bsp98I	GGATCC	GGATCC	
Bsp100I	GGWCC	GGWCC	

Bsp101I	TTCGAA	TTCGAA	
Bsp102I	TTCGAA	TTCGAA	
Bsp103I	CCWGG	CCWGG	
Bsp104I	TTCGAA	TTCGAA	
Bsp105I	GATC	GATC	
Bsp106I	ATCGAT	ATCGAT	
M.Bsp106I	ATCGAT	ATCGAT	
Bsp107I	CTGCAG	CTGCAG	
Bsp108I	CTGCAG	CTGCAG	
Bsp116I	CCGG	CCGG	
Bsp117I	GRGCYC	GRGCYC	
Bsp119I	TTCGAA	TTCGAA	F.
Bsp120I	GGGCCC	GGGCCC	FG.
Bsp121I	GCATGC	GCATGC	
Bsp122I	GATC	GATC	
Bsp123I	CGCG	CGCG	
Bsp125I	ATCGAT	ATCGAT	
Bsp126I	ATCGAT	ATCGAT	
Bsp127I	ATCGAT	ATCGAT	
Bsp128I	GGWCC	GGWCC	
Bsp129I	CTCGAG	CTCGAG	
Bsp130I	GGATCC	GGATCC	
Bsp131I	GGATCC	GGATCC	
Bsp132I	GGWCC	GGWCC	
Bsp133I	GGWCC	GGWCC	
Bsp135I	GATC	GATC	
Bsp136I	GATC	GATC	
Bsp137I	GGCC	GGCC	
Bsp138I	GATC	GATC	
Bsp139I	CTCGAG	CTCGAG	
Bsp140I	CTCGAG	CTCGAG	
Bsp141I	CTCGAG	CTCGAG	
Bsp142I	CTCGAG	CTCGAG	
Bsp143I	GATC	GATC	F.
Bsp143II	RGCGCY	RGCGCY	F.
M.Bsp143II	RGCGCY	RGCGCY	
Bsp144I	GGATCC	GGATCC	
Bsp145I	ATCGAT	ATCGAT	
Bsp146I	GTGCAC	GTGCAC	
Bsp147I	GATC	GATC	
Bsp148I	TTCGAA	TTCGAA	
Bsp151I	TTCGAA	TTCGAA	
Bsp211I	GGCC	GGCC	
Bsp226I	GGCC	GGCC	
Bsp228I	TCCGGA	TCCGGA	
Bsp233I	TCCGGA	TCCGGA	
Bsp241I	TTCGAA	TTCGAA	
Bsp268I	CTGCAG	CTGCAG	
Bsp317I	CCWGG	CCWGG	
Bsp423I	GCAGC	GCTGC	
Bsp508I	TCCGGA	TCCGGA	
Bsp519I	GRGCYC	GRGCYC	
Bsp548I	CCNGG	CCNGG	
Bsp774I	?	?	
Bsp881I	GGCC	GGCC	
Bsp1260I	GGWCC	GGWCC	
Bsp1261I	GGCC	GGCC	
Bsp1286I	GDGCHC	GDGCHC	JKNR.
Bsp1407I	TGTACA	TGTACA	FK.
Bsp1566I	?	?	
Bsp1591I	GGTNACC	GGTNACC	
Bsp1591II	CCGG	CCGG	
Bsp1593I	GGCC	GGCC	
Bsp1720I	GCTNAGC	GCTNAGC	IV.
Bsp1883I	?	?	
Bsp1894I	GGNCC	GGNCC	
Bsp2013I	GGCC	GGCC	
Bsp2095I	GATC	GATC	
Bsp2362I	GGCC	GGCC	
Bsp2500I	GGCC	GGCC	
Bsp4009I	GGATCC	GGATCC	
Bsp9002I	?	?	
BspAI	GATC	GATC	
BspA2I	CCTAGG	CCTAGG	
Bsp153AI	CAGCTG	CAGCTG	
BspAAI	CTCGAG	CTCGAG	
BspAAII	TCTAGA	TCTAGA	
BspAAIII	GGATCC	GGATCC	
BspACI	CCGC	GCGG	I.
BspANI	GGCC	GGCC	X.

BspBI	CTGCAG	CTGCAG	
BspBII	GGNCC	GGNCC	
BspB2I	?	?	
BspBDG2I	GGCC	GGCC	
BspBRI	GGCC	GGCC	
BspBS31I	GAAGAC	GTCTTC	
BspBSE18I	GGCC	GGCC	
BspBake1I	GGCC	GGCC	
BspCI	CGATCG	CGATCG	
BspCHE15I	GGCC	GGCC	
BspCNI	CTCAG	CTGAG	N.
M.BspCNI	CTCAG	CTCAG	
BspDI	ATCGAT	ATCGAT	N.
BspD6II	CTGAAG	CTTCAG	
BspD6III	?	?	
BspEI	TCCGGA	TCCGGA	N.
M.BspEI	TCCGGA	TCCGGA	
BspFI	GATC	GATC	
BspF4I	GGNCC	GGNCC	
BspF53I	GGWCC	GGWCC	
BspF105I	CCSGG	CCSGG	
BspGI	CTGGAC	GTCCAG	
BspGHA1I	GGCC	GGCC	
BspHI	TCATGA	TCATGA	N.
M.BspHI	TCATGA	TCATGA	
BspH22I	TTCGAA	TTCGAA	
BspH43I	CCWGG	CCWGG	
BspH103I	TTCGAA	TTCGAA	
BspH106I	TTCGAA	TTCGAA	
BspH106II	GGCC	GGCC	
BspH226I	TCCGGA	TCCGGA	
BspIAB59I	?	?	
BspIS4I	GAAGAC	GTCTTC	
M.BspIS4I	GAAGAC	GAAGAC	
BspJI	GATC	GATC	
BspJII	ATCGAT	ATCGAT	
BspJ64I	GATC	GATC	
BspJ67I	CCSGG	CCSGG	
BspJ74I	CTGGAG	CTCCAG	
BspJ76I	CGCG	CGCG	
BspJ105I	GGWCC	GGWCC	
BspJ106I	GGTACC	GGTACC	
BspKI	GGCC	GGCC	
BspKT5I	CTGAAG	CTTCAG	
BspKT6I	GATC	GATC	
M.BspKT6I	GATC	GATC	
BspKT8I	AAGCTT	AAGCTT	
BspK1aI	?	?	
BspLI	GGNNCC	GGNNCC	F.
BspLAI	GCGC	GCGC	
BspLAII	TTCGAA	TTCGAA	
BspLAIII	AAGCTT	AAGCTT	
BspLRI	GGCC	GGCC	
BspLS2I	GDGCHC	GDGCHC	
BspLU4I	CYCGRG	CYCGRG	
BspLU11I	ACATGT	ACATGT	M.
BspLU11II	TCTAGA	TCTAGA	
BspLU11III	GGGAC	GTCCC	
M1.BspLU11III	GGGAC	GGGAC	
M2.BspLU11III	GGGAC	GGGAC	
BspMI	ACCTGC	GCAGGT	N.
M1.BspMI	ACCTGC	ACCTGC	
M2.BspMI	ACCTGC	ACCTGC	
BspMII	TCCGGA	TCCGGA	
M.BspMII	TCCGGA	TCCGGA	
BspM39I	CAGCTG	CAGCTG	
BspM90I	GTATAC	GTATAC	
BspMAI	CTGCAG	CTGCAG	X.
BspMKI	GTCGAC	GTCGAC	
BspNI	CCWGG	CCWGG	
BspNCI	CCAGA	TCTGG	
BspO4I	CAGCTG	CAGCTG	
BspOVI	GACNNNNNGTC	GACNNNNNGTC	
BspOVII	ATCGAT	ATCGAT	
BspPI	GGATC	GATCC	F.
BspPR1I	?	?	
BspQI	GCTCTTC	GAAGAGC	
BspRI	GGCC	GGCC	
M.BspRI	GGCC	GGCC	
BspR7I	CCTNAGG	CCTNAGG	

BspSI	CCWGG	CCWGG	
BspS122I	CTGCAG	CTGCAG	
BspSSI	?	?	
BspST5I	GCATC	GATGC	
M.BspST5I	GCATC	GCATC	
BspTI	CTTAAG	CTTAAG	F.
BspT104I	TTCGAA	TTCGAA	K.
BspT107I	GGYRCC	GGYRCC	K.
BspTNI	GGTCTC	GAGACC	X.
BspTS514I	GAAGAC	GTCTTC	
BspUI	GCSGC	GCSGC	
BspVI	GAAGAC	GTCTTC	
BspWI	GCNNNNNNNGC	GCNNNNNNNGC	
BspXI	ATCGAT	ATCGAT	G.
BspXII	TGATCA	TGATCA	
BspZEI	ATCGAT	ATCGAT	
BsrI	ACTGG	CCAGT	N.
M1.BsrI	ACTGG	ACTGG	
M2.BsrI	ACTGG	ACTGG	
BsrAI	GGWCC	GGWCC	
BsrBI	CCGCTC	GAGCGG	N.
M1.BsrBI	CCGCTC	CCGCTC	
M2.BsrBI	CCGCTC	CCGCTC	
BsrBRI	GATNNNNATC	GATNNNNATC	
BsrCI	ATCGAT	ATCGAT	
BsrDI	GCAATG	CATTGC	N.
BsrEI	CTCTTC	GAAGAG	
BsrFI	RCCGGY	RCCGGY	N.
M.BsrFI	RCCGGY	RCCGGY	
BsrGI	TGTACA	TGTACA	N.
M.BsrGI	?	?	
BsrGII	?	?	
BsrHI	GCGCGC	GCGCGC	
BsrMI	GATC	GATC	
BsrPI	?	?	
BsrPII	GATC	GATC	
BsrSI	ACTGG	CCAGT	R.
BsrVI	GCAGC	GCTGC	
BsrWI	GGATC	GATCC	
BsrXI	TCTAGA	TCTAGA	
BssI	GGNNCC	GGNNCC	
BssAI	RCCGGY	RCCGGY	C.
BssBI	GCGCGC	GCGCGC	
BssCI	GGCC	GGCC	
BssECI	CCNNGG	CCNNGG	I.
BssFI	GCNGC	GCNGC	
BssGI	CCANNNNNNTGG	CCANNNNNNTGG	
BssGII	GATC	GATC	
BssHI	CTCGAG	CTCGAG	
M.BssHI	CTCGAG	CTCGAG	
BssHII	GCGCGC	GCGCGC	AJKMNOQRSX.
M.BssHII	GCGCGC	GCGCGC	
BssIMI	GGGTC	GACCC	
BssKI	CCNGG	CCNGG	N.
BssMI	GATC	GATC	V.
BssNI	GRCGYC	GRCGYC	V.
BssNAI	GTATAC	GTATAC	IV.
BssPI	?	?	
BssSI	CACGAG	CTCGTG	N.
M.BssSI	CACGAG	CACGAG	
BssT1I	CCWWGG	CCWWGG	IV.
BssXI	GCNGC	GCNGC	
BstI	GGATCC	GGATCC	
M.BstI	GGATCC	GGATCC	
Bst1I	CCWGG	CCWGG	
Bst2I	CCWGG	CCWGG	
Bst6I	CTCTTC	GAAGAG	IV.
Bst11I	ACTGG	CCAGT	
Bst12I	GCAGC	GCTGC	
Bst16I	RGCGCY	RGCGCY	
Bst19I	GCATC	GATGC	
Bst19II	GATC	GATC	
Bst22I	CCNNNNNNNGG	CCNNNNNNNGG	
Bst28I	ATCGAT	ATCGAT	
Bst29I	CCTNAGG	CCTNAGG	
Bst30I	CCTNAGG	CCTNAGG	
Bst31I	GGTNACC	GGTNACC	
Bst38I	CCWGG	CCWGG	
Bst40I	CCGG	CCGG	
Bst44I	?	?	

Bst71I	GCAGC	GCTGC	
Bst77I	TGATCA	TGATCA	
Bst98I	CTTAAG	CTTAAG	R.
Bst100I	CCWGG	CCWGG	
Bst158I	CTCTTC	GAAGAG	
Bst170I	TGTACA	TGTACA	
Bst170II	AAGCTT	AAGCTT	
Bst224I	CCWWGG	CCWWGG	
Bst295I	CTNAG	CTNAG	
Bst1107I	GTATAC	GTATAC	FKM.
Bst1126I	GGATCC	GGATCC	
Bst1274I	GATC	GATC	
Bst1473I	WCCGGW	WCCGGW	
Bst1473II	RGCGCY	RGCGCY	
Bst2464I	GGATCC	GGATCC	
Bst2902I	GGATCC	GGATCC	
BstAI	?	?	
BstACI	GRCGYC	GRCGYC	I.
BstAPI	GCANNNNNTGC	GCANNNNNTGC	IN.
BstAUI	TGTACA	TGTACA	IV.
BstBI	TTCGAA	TTCGAA	N.
Bst2BI	CACGAG	CTCGTG	IV.
BstBAI	YACGTR	YACGTR	IV.
BstBAII	CYCGRG	CYCGRG	
BstBSI	GTATAC	GTATAC	
BstB7SI	RCCGGY	RCCGGY	
BstBS32I	GAAGAC	GTCTTC	
BstBZ153I	GCGCGC	GCGCGC	
BstCI	GGCC	GGCC	
Bst4CI	ACNGT	ACNGT	IV.
BstC8I	GCNNGC	GCNNGC	I.
BstDI	GGTNACC	GGTNACC	
BstD102I	CCGCTC	GAGCGG	
BstDEI	CTNAG	CTNAG	IV.
BstDSI	CCRYGG	CCRYGG	IV.
BstDZ247I	CCCGT	ACGGG	
BstEI	?	?	
BstEII	GGTNACC	GGTNACC	GHJMNORSU.
M.BstEII	GGTNACC	GGTNACC	
BstEIII	GATC	GATC	
M.BstEIII	GATC	GATC	
BstENI	CCTNNNNNAGG	CCTNNNNNAGG	IV.
BstENII	GATC	GATC	
BstEZ359I	GTTAAC	GTTAAC	
BstFI	AAGCTT	AAGCTT	
BstF5I	GGATG	CATCC	INV.
M1.BstF5I	GGATG	GGATG	
M2.BstF5I	GGATG	GGATG	
M3.BstF5I	GGATG	GGATG	
M4.BstF5I	GGATG	GGATG	
BstFNI	CGCG	CGCG	IV.
BstFZ438I	CCCGC	GCGGG	
BstGI	TGATCA	TGATCA	
BstGII	CCWGG	CCWGG	
M.BstGII	CCWGG	CCWGG	
BstGZ53I	CGTCTC	GAGACG	
BstHI	CTCGAG	CTCGAG	
BstH2I	RGCGCY	RGCGCY	IV.
BstH9I	GGATC	GATCC	
BstHHI	GCGC	GCGC	IV.
BstHPI	GTTAAC	GTTAAC	
BstHZ55I	CCANNNNNNTGG	CCANNNNNNTGG	
BstIZ316I	CACNNNGTG	CACNNNGTG	
BstJI	GGCC	GGCC	
BstJZ301I	CTNAG	CTNAG	
BstKI	TGATCA	TGATCA	
BstKTI	GATC	GATC	I.
BstKZ418I	?	?	
BstLI	CTCGAG	CTCGAG	
BstLVI	ATCGAT	ATCGAT	
M.BstLVI	ATCGAT	ATCGAT	
BstMI	AGTACT	AGTACT	
BstM6I	CCWGG	CCWGG	
BstMAI	GTCTC	GAGAC	IV.
BstMBI	GATC	GATC	IV.
BstMCI	CGRYCG	CGRYCG	IV.
BstMWI	GCNNNNNNNGC	GCNNNNNNNGC	I.
BstMZ611I	CCNGG	CCNGG	
BstNI	CCWGG	CCWGG	N.
M.BstNI	CCWGG	CCWGG	

Bst31NI	CCGCTC	GAGCGG	
M.BstNBI	GASTC	GASTC	
M.BstNBII	?	?	
BstNSI	RCATGY	RCATGY	IV.
BstNSII	CYCGRG	CYCGRG	
BstNZ169I	ATCGAT	ATCGAT	
BstOI	CCWGG	CCWGG	R.
BstOZ616I	GGGAC	GTCCC	
BstPI	GGTNACC	GGTNACC	K.
BstPAI	GACNNNNGTC	GACNNNNGTC	IV.
BstPZ740I	CTTAAG	CTTAAG	
BstQI	GGATCC	GGATCC	
Bst4QI	GGWCC	GGWCC	
Bst7QI	CYCGRG	CYCGRG	
Bst7QII	CCWGG	CCWGG	
BstRI	GATATC	GATATC	
BstRZ246I	ATTTAAAT	ATTTAAAT	
BstRZ459I	?	?	
BstSI	CYCGRG	CYCGRG	
BstSCI	CCNGG	CCNGG	I.
M1.BstSEI	GAGTC	GAGTC	
M2.BstSEI	GAGTC	GAGTC	
BstSFI	CTRYAG	CTRYAG	I.
BstSNI	TACGTA	TACGTA	IV.
BstSWI	ATTTAAAT	ATTTAAAT	
BstTI	CCANNNNNNTGG	CCANNNNNNTGG	
BstT7I	TGATCA	TGATCA	
BstT9I	GGTNACC	GGTNACC	
BstT10I	GGTNACC	GGTNACC	
Bst31TI	GGATC	GATCC	
BstTS5I	GAAGAC	GTCTTC	
BstUI	CGCG	CGCG	N.
Bst2UI	CCWGG	CCWGG	IV.
BstVI	CTCGAG	CTCGAG	
M.BstVI	CTCGAG	CTCGAG	
BstV1I	GCAGC	GCTGC	I.
BstV2I	GAAGAC	GTCTTC	IV.
BstWI	CCTNNNNNAGG	CCTNNNNNAGG	
BstXI	CCANNNNNNTGG	CCANNNNNNTGG	AFGHIJKMNOQRVX.
M.BstXI	CCANNNNNNTGG	CCANNNNNNTGG	
BstXII	GATC	GATC	
BstX2I	RGATCY	RGATCY	IV.
BstYI	RGATCY	RGATCY	N.
M.BstYI	RGATCY	RGATCY	
BstZI	CGGCCG	CGGCCG	R.
BstZ1I	TCCGGA	TCCGGA	
BstZ1II	AAGCTT	AAGCTT	
M.BstZ1II	AAGCTT	AAGCTT	
BstZ2I	GACNNNNNGTC	GACNNNNNGTC	
BstZ3I	TCCGGA	TCCGGA	
BstZ4I	CYCGRG	CYCGRG	
BstZ5I	CGRYCG	CGRYCG	
BstZ6I	CCTNAGG	CCTNAGG	
BstZ7I	GRGCYC	GRGCYC	
BstZ8I	CGATCG	CGATCG	
BstZ9I	ACGCGT	ACGCGT	
BstZ10I	CCNNGG	CCNNGG	
BstZ10II	TGATCA	TGATCA	
BstZ12I	?	?	
BstZ13I	?	?	
BstZ14I	?	?	
BstZ15I	GDGCHC	GDGCHC	
BstZ16I	GTCGAC	GTCGAC	
BstZ17I	GTATAC	GTATAC	N.
Bsu6I	CTCTTC	GAAGAG	
Bsu15I	ATCGAT	ATCGAT	F.
M.Bsu15I	ATCGAT	ATCGAT	
Bsu22I	TCCGGA	TCCGGA	
Bsu23I	TCCGGA	TCCGGA	
Bsu36I	CCTNAGG	CCTNAGG	NR.
M.Bsu36I	CCTNAGG	CCTNAGG	
Bsu54I	GGNCC	GGNCC	
Bsu90I	GGATCC	GGATCC	
Bsu121I	?	?	
Bsu1076I	GGCC	GGCC	
Bsu1114I	GGCC	GGCC	
Bsu1145I	?	?	
Bsu1192I	CCGG	CCGG	
Bsu1192II	CGCG	CGCG	
Bsu1193I	CGCG	CGCG	

Bsu1259I	?	?	
Bsu1532I	CGCG	CGCG	
Bsu1854I	GRGCYC	GRGCYC	
Bsu2413I	?	?	
Bsu5044I	GGNCC	GGNCC	
Bsu6633I	CGCG	CGCG	
M.Bsu6633I	CGCG	CGCG	
Bsu8565I	GGATCC	GGATCC	
Bsu8646I	GGATCC	GGATCC	
BsuBI	CTGCAG	CTGCAG	
M.BsuBI	CTGCAG	CTGCAG	
BsuB519I	GGATCC	GGATCC	
BsuB763I	GGATCC	GGATCC	
BsuCI	?	?	
M.BsuCI	?	?	
BsuEII	CGCG	CGCG	
M.BsuEII	CGCG	CGCG	
BsuFI	CCGG	CCGG	
M.BsuFI	CCGG	CCGG	
BsuF2I	?	?	
BsuMI	CTCGAG	CTCGAG	
M1.BsuMI	?	?	
M2.BsuMI	?	?	
BsuRI	GGCC	GGCC	FI.
M.BsuRI	GGCC	GGCC	
BsuTUI	ATCGAT	ATCGAT	X.
BsxI	ACTGGG	CCCAGT	
BtcI	GATC	GATC	
BteI	GGCC	GGCC	
BtgI	CCRYGG	CCRYGG	N.
BtgAI	GTCGAC	GTCGAC	
BtgAII	GCATGC	GCATGC	
BtgZI	GCGATG	CATCGC	N.
BthI	CTCGAG	CTCGAG	
BthII	GGATC	GATCC	
Bth84I	GATC	GATC	
Bth211I	GATC	GATC	
Bth213I	GATC	GATC	
Bth221I	GATC	GATC	
Bth617I	GGATC	GATCC	
Bth945I	GATC	GATC	
Bth1140I	GATC	GATC	
Bth1141I	GATC	GATC	
Bth1202I	ATCGAT	ATCGAT	
Bth1786I	GATC	GATC	
Bth1795I	CTGGAG	CTCCAG	
Bth1997I	GATC	GATC	
Bth2350I	CAGCTG	CAGCTG	
Bth9411I	CTGCAG	CTGCAG	
Bth9415I	ATCGAT	ATCGAT	
BthAI	GGWCC	GGWCC	
BthCI	GCNGC	GCNGC	
BthCanI	GATC	GATC	
BthDI	CCWGG	CCWGG	
BthEI	CCWGG	CCWGG	
M.BthIPS78	ACGGC	ACGGC	
BthP35I	CTRYAG	CTRYAG	
BtiI	GGWCC	GGWCC	
BtkI	CGCG	CGCG	
BtkII	GATC	GATC	
BtrI	CACGTC	GACGTG	IV.
BtsI	GCAGTG	CACTGC	N.
M1.BtsI	GCAGTG	GCAGTG	
M2.BtsI	GCAGTG	GCAGTG	
BtsCI	GGATG	CATCC	N.
M.BtsCI	GGATG	GGATG	
BtsPI	GGGTC	GACCC	
BtuI	ATCGAT	ATCGAT	
Btu33I	GATC	GATC	
Btu34I	GATC	GATC	
Btu34II	RGCGCY	RGCGCY	
Btu36I	GATC	GATC	
Btu37I	GATC	GATC	
Btu39I	GATC	GATC	
Btu41I	GATC	GATC	
BtuMI	TCGCGA	TCGCGA	V.
BveI	ACCTGC	GCAGGT	F.
BvuI	GRGCYC	GRGCYC	
BvuBI	CGTACG	CGTACG	
CacI	GATC	GATC	

Cac8I	GCNNGC	GCNNGC	N.
M.Cac8I	GCNNGC	GCNNGC	
Cac824I	GCNGC	GCNGC	
M.Cac824I	GCNGC	GCNGC	
CaiI	CAGNNNCTG	CAGNNNCTG	F.
CalI	?	?	
Cas2I	CGATCG	CGATCG	
CauI	GGWCC	GGWCC	
CauII	CCSGG	CCSGG	
CauIII	CTGCAG	CTGCAG	
CauB3I	TCCGGA	TCCGGA	
CbiI	TTCGAA	TTCGAA	
CboI	CCGG	CCGG	
M.CboI	CCGG	CCGG	
CbrI	CCWGG	CCWGG	
CceI	CCGG	CCGG	
CciNI	GCGGCCGC	GCGGCCGC	IV.
CcoI	GCCGGC	GCCGGC	
CcoP31I	GATC	GATC	
CcoP73I	GTAC	GTAC	
CcoP76I	GATC	GATC	
CcoP84I	GATC	GATC	
CcoP95I	GCGC	GCGC	
CcoP95II	GATC	GATC	
CcoP215I	GCNGC	GCNGC	
CcoP216I	GCNGC	GCNGC	
CcoP219I	GATC	GATC	
CcrI	CTCGAG	CTCGAG	
M.CcrMI	GANTC	GANTC	
CcuI	GGNCC	GGNCC	
CcyI	GATC	GATC	
CdiI	CATCG	CGATG	
M.CdiI	TGGCCA	TGGCCA	
Cdi27I	CCWGG	CCWGG	
M.Cdi630I	TGGCCA	TGGCCA	
M.Cdi630II	?	?	
M.Cdi630III	CCSSGG	CCSSGG	
M.Cdi630IV	GCWGC	GCWGC	
Cdi630V	?	?	
CdiAI	GGNCC	GGNCC	
CdiCD6I	GGNCC	GGNCC	
M.CdiCD6I	GGNCC	GGNCC	
CdiCD6II	GATC	GATC	
M.CdiCD6II	GATC	GATC	
CelI	GGATCC	GGATCC	
CelII	GCTNAGC	GCTNAGC	M.
CeqI	GATATC	GATATC	
M.CeqI	GATATC	GATATC	
I-CeuI	CGTAACTATAACGGTCCTAAGGTAGCGAA	TTCGCTACCTTAGGACCGTTATAGTTACG	N.
CfaI	RAATTY	RAATTY	
CflI	CTGCAG	CTGCAG	
CfoI	GCGC	GCGC	GMRS.
CfrI	YGGCCR	YGGCCR	F.
M.CfrI	YGGCCR	YGGCCR	
Cfr4I	GGNCC	GGNCC	
Cfr5I	CCWGG	CCWGG	
Cfr6I	CAGCTG	CAGCTG	
M.Cfr6I	CAGCTG	CAGCTG	
Cfr7I	GGTNACC	GGTNACC	
Cfr8I	GGNCC	GGNCC	
Cfr9I	CCCGGG	CCCGGG	FO.
M.Cfr9I	CCCGGG	CCCGGG	
Cfr10I	RCCGGY	RCCGGY	FGKO.
M.Cfr10I	RCCGGY	RCCGGY	
Cfr11I	CCWGG	CCWGG	
Cfr13I	GGNCC	GGNCC	AFKO.
M.Cfr13I	GGNCC	GGNCC	
Cfr14I	YGGCCR	YGGCCR	
Cfr19I	GGTNACC	GGTNACC	
Cfr20I	CCWGG	CCWGG	
Cfr22I	CCWGG	CCWGG	
Cfr23I	GGNCC	GGNCC	
Cfr24I	CCWGG	CCWGG	
Cfr25I	CCWGG	CCWGG	
Cfr27I	CCWGG	CCWGG	
Cfr28I	CCWGG	CCWGG	
Cfr29I	CCWGG	CCWGG	
Cfr30I	CCWGG	CCWGG	
Cfr31I	CCWGG	CCWGG	
Cfr32I	AAGCTT	AAGCTT	

Cfr33I	GGNCC	GGNCC	
Cfr35I	CCWGG	CCWGG	
Cfr37I	CCGCGG	CCGCGG	
Cfr38I	YGGCCR	YGGCCR	
Cfr39I	YGGCCR	YGGCCR	
Cfr40I	YGGCCR	YGGCCR	
Cfr41I	CCGCGG	CCGCGG	
Cfr42I	CCGCGG	CCGCGG	F.
M.Cfr42I	CCGCGG	CCGCGG	
Cfr43I	CCGCGG	CCGCGG	
Cfr45I	GGNCC	GGNCC	
Cfr45II	CCGCGG	CCGCGG	
Cfr46I	GGNCC	GGNCC	
Cfr47I	GGNCC	GGNCC	
Cfr48I	GRGCYC	GRGCYC	
Cfr51I	CGATCG	CGATCG	
Cfr52I	GGNCC	GGNCC	
Cfr54I	GGNCC	GGNCC	
Cfr55I	YGGCCR	YGGCCR	
Cfr56I	GGTCTC	GAGACC	
Cfr57I	TCCGGA	TCCGGA	
Cfr58I	CCWGG	CCWGG	
Cfr59I	YGGCCR	YGGCCR	
Cfr92I	CTTAAG	CTTAAG	
CfrAI	GCANNNNNNNNGTGG	CCACNNNNNNNNTGC	
M.CfrAI	GCANNNNNNNNGTGG	GCANNNNNNNNGTGG	
CfrA4I	CTGCAG	CTGCAG	
CfrBI	CCWWGG	CCWWGG	
M.CfrBI	CCWWGG	CCWWGG	
CfrJ4I	CCCGGG	CCCGGG	
CfrJ5I	GCGCGC	GCGCGC	
CfrNI	GGNCC	GGNCC	
CfrS37I	CCWGG	CCWGG	
CfuI	GATC	GATC	
CfuII	CTGCAG	CTGCAG	
M.CfuIII	?	?	
CglI	GCSGC	GCSGC	
M.CglI	GCSGC	GCSGC	
Cgl165I	?	?	
CglAI	GCATGC	GCATGC	
CglAII	GTCGAC	GTCGAC	
M.CglASI	GCSGC	GCSGC	
ChaI	GATC	GATC	
ChiI	?	?	
ChuI	AAGCTT	AAGCTT	
I-ChuI	GAAGGTTTGGCACCTCGATGTCGGCTCATC	GATGAGCCGACATCGAGGTGCCAAACCTTC	
ChuII	GTYRAC	GTYRAC	
ChyI	AGGCCT	AGGCCT	
Cin1467I	GATC	GATC	
CjaI	CTCGAG	CTCGAG	
CjeI	CCANNNNNNGT	ACNNNNNNTGG	
CjeI	ACNNNNNNTGG	CCANNNNNNGT	
M.CjeNI	GAATTC	GAATTC	
CjeNII	GAGNNNNNGT	ACNNNNNCTC	
CjePI	CCANNNNNNNTC	GANNNNNNNTGG	
CjePI	GANNNNNNNTGG	CCANNNNNNNTC	
CjeP338I	GATC	GATC	
CjeP338II	GCATC	GATGC	
CjuI	CAYNNNNNRTG	CAYNNNNNRTG	
CjuII	CAYNNNNNCTC	GAGNNNNNRTG	
ClaI	ATCGAT	ATCGAT	ABHKMNRSU.
M.ClaI	ATCGAT	ATCGAT	K.
ClcI	CTGCAG	CTGCAG	
ClcII	TGCGCA	TGCGCA	
CliI	GGWCC	GGWCC	
CliII	TGCGCA	TGCGCA	
CliIII	?	?	
ClmI	GGCC	GGCC	
ClmII	GGWCC	GGWCC	
CltI	GGCC	GGCC	
CluI	?	?	
I-CmoeI	TCGTAGCAGCTCACGGTT	AACCGTGAGCTGCTACGA	
CpaI	GATC	GATC	
I-CpaI	CGATCCTAAGGTAGCGAAATTCA	TGAATTTCGCTACCTTAGGATCG	
I-CpaII	CCCGGCTAACTCTGTGCCAG	CTGGCACAGAGTTAGCCGGG	
Cpa1150I	CGCG	CGCG	
CpaAI	CGCG	CGCG	
CpeI	TGATCA	TGATCA	
CpfI	GATC	GATC	
CpfAI	GATC	GATC	

CpoI	CGGWCCG	CGGWCCG	AFK.
CprJK699I	?	?	
CprJK722I	ATTAAT	ATTAAT	
I-CreI	CTGGGTTCAAAACGTCGTGAGACAGTTTGG	CCAAACTGTCTCACGACGTTTTGAACCCAG	
I-CreII	TGTAGCTGCTCATGGTT	AACCATGAGCAGCTACA	
M.CreDnmt1	?	?	
CscI	CCGCGG	CCGCGG	
CseI	GACGC	GCGTC	F.
CsiAI	ACCGGT	ACCGGT	
CsiBI	GCGGCCGC	GCGGCCGC	
I-CsmI	GTACTAGCATGGGGTCAAATGTCTTTCTGG	CCAGAAAGACATTTGACCCCATGCTAGTAC	
CspI	CGGWCCG	CGGWCCG	OR.
Csp2I	GGCC	GGCC	
Csp4I	ATCGAT	ATCGAT	
Csp5I	GATC	GATC	
Csp6I	GTAC	GTAC	F.
M.Csp6I	GTAC	GTAC	
Csp45I	TTCGAA	TTCGAA	OR.
Csp231I	AAGCTT	AAGCTT	
M.Csp231I	AAGCTT	AAGCTT	
Csp1470I	GCGC	GCGC	
CspAI	ACCGGT	ACCGGT	C.
CspBI	GCGGCCGC	GCGGCCGC	
CspCI	CAANNNNNGTGG	CCACNNNNNTTG	N.
CspCI	CCACNNNNNTTG	CAANNNNNGTGG	N.
Csp68KI	GGWCC	GGWCC	
M.Csp68KI	GGWCC	GGWCC	
Csp68KII	TTCGAA	TTCGAA	
Csp68KIII	ATGCAT	ATGCAT	
M.Csp68KIV	CCGG	CCGG	
M.Csp68KV	GGCC	GGCC	
Csp68KVI	CGCG	CGCG	
CspKVI	CGCG	CGCG	
CstI	CTGCAG	CTGCAG	
CstMI	AAGGAG	CTCCTT	
CsuI	?	?	
Cte1I	CCGCGG	CCGCGG	
Cte1179I	GATC	GATC	
Cte1180I	GATC	GATC	
CthI	TGATCA	TGATCA	
CthII	CCWGG	CCWGG	
CtyI	GATC	GATC	
M.CvaI	?	?	
CveI	?	?	
CviI	?	?	
CviAI	GATC	GATC	
M.CviAI	GATC	GATC	
CviAII	CATG	CATG	N.
M.CviAII	CATG	CATG	
M.CviAIV	RGCB	RGCB	
M.CviAV	?	?	
CviBI	GANTC	GANTC	
M.CviBI	GANTC	GANTC	
M.CviBII	GATC	GATC	
M.CviBIII	TCGA	TCGA	
CviCI	GANTC	GANTC	
CviDI	GANTC	GANTC	
CviEI	GANTC	GANTC	
CviFI	GANTC	GANTC	
CviGI	GANTC	GANTC	
CviHI	GATC	GATC	
CviJI	RGCY	RGCY	VX.
M.CviJI	RGCB	RGCB	
CviKI	RGCY	RGCY	
CviKI-1	RGCY	RGCY	N.
M.CviKI	RGCY	RGCY	
CviLI	RGCY	RGCY	
CviMI	RGCY	RGCY	
CviNI	RGCY	RGCY	
CviOI	RGCY	RGCY	
M.CviPI	GC	GC	
M.CviPII	?	?	
CviQI	GTAC	GTAC	
M.CviQI	GTAC	GTAC	
M.CviQII	RAR	RAR	
M.CviQIII	TCGA	TCGA	
M.CviQVI	GANTC	GANTC	
M.CviQVII	CATG	CATG	
CviRI	TGCA	TGCA	
M.CviRI	TGCA	TGCA	

CviRII	GTAC	GTAC	
M.CviRII	GTAC	GTAC	
M.CviSI	TGCA	TGCA	
M.CviSII	CATG	CATG	
CviSIII	TCGA	TCGA	
M.CviSIII	TCGA	TCGA	
CvnI	CCTNAGG	CCTNAGG	
I-CvuI	CTGGGTTCAAAACGTCGTGAGACAGTTTGG	CCAAACTGTCTCACGACGTTTTGAACCCAG	
DaqI	GTGCAC	GTGCAC	
M.DcaI	?	?	
M.DcaII	?	?	
DdeI	CTNAG	CTNAG	BGMNORS.
M.DdeI	CTNAG	CTNAG	
DdeII	CTCGAG	CTCGAG	
I-DdiI	TTTTTTGGTCATCCAGAAGTATAT	ATATACTTCTGGATGACCAAAAAA	
DdsI	GGATCC	GGATCC	
M.DhaYORF2200	TGGCCA	TGGCCA	
DinI	GGCGCC	GGCGCC	V.
I-DirI	?	?	
DmaI	CAGCTG	CAGCTG	
DmoI	?	?	
I-DmoI	ATGCCTTGCCGGGTAAGTTCCGGCGCGCAT	ATGCGCGCCGGAACTTACCCGGCAAGGCAT	
DpaI	AGTACT	AGTACT	
DpnI	GATC	GATC	BEFGMNRS.
DpnII	GATC	GATC	N.
M1.DpnII	GATC	GATC	
M2.DpnII	GATC	GATC	
DraI	TTTAAA	TTTAAA	ABFGIJKMNOQRSUVXY.
M.DraI	TTTAAA	TTTAAA	
DraII	RGGNCCY	RGGNCCY	GM.
M.DraII	RGGNCCY	RGGNCCY	
DraIII	CACNNNGTG	CACNNNGTG	GIMNV.
M.DraIII	CACNNNGTG	CACNNNGTG	
DrdI	GACNNNNNNGTC	GACNNNNNNGTC	N.
DrdII	GAACCA	TGGTTC	
DrdIII	CGATCG	CGATCG	
DrdAI	CCGCGG	CCGCGG	
DrdBI	CCGCGG	CCGCGG	
DrdCI	CCGCGG	CCGCGG	
DrdDI	CTCGAG	CTCGAG	
DrdEI	CCGCGG	CCGCGG	
DrdFI	CCGCGG	CCGCGG	
H-DreI	CAAAACGTCGTAAGTTCCGGCGCG	CGCGCCGGAACTTACGACGTTTTG	
DriI	GACNNNNNGTC	GACNNNNNGTC	I.
DsaI	CCRYGG	CCRYGG	
DsaII	GGCC	GGCC	
DsaIII	RGATCY	RGATCY	
DsaIV	GGWCC	GGWCC	
DsaV	CCNGG	CCNGG	
M.DsaV	CCNGG	CCNGG	
DsaVI	GTMKAC	GTMKAC	
DseDI	GACNNNNNNGTC	GACNNNNNNGTC	I.
Dsp1I	CCGCGG	CCGCGG	
EacI	GGATC	GATCC	
M.EacI	GGATC	GGATC	
EaeI	YGGCCR	YGGCCR	AKMN.
M.EaeI	YGGCCR	YGGCCR	
Eae2I	CTCGAG	CTCGAG	
Eae46I	CCGCGG	CCGCGG	
EaeAI	CCCGGG	CCCGGG	
EaePI	CTGCAG	CTGCAG	
EagI	CGGCCG	CGGCCG	GN.
M.EagI	CGGCCG	CGGCCG	
EagBI	CGATCG	CGATCG	
EagKI	CCWGG	CCWGG	
EagMI	GGWCC	GGWCC	
Eam1104I	CTCTTC	GAAGAG	F.
Eam1105I	GACNNNNNGTC	GACNNNNNGTC	FK.
EarI	CTCTTC	GAAGAG	N.
M1.EarI	CTCTTC	CTCTTC	
M2.EarI	CTCTTC	CTCTTC	
EcaI	GGTNACC	GGTNACC	
M.EcaI	GGTNACC	GGTNACC	
EcaII	CCWGG	CCWGG	
EccI	CCGCGG	CCGCGG	
EciI	GGCGGA	TCCGCC	N.
Eci125I	GGTNACC	GGTNACC	
EciAI	TACGTA	TACGTA	
EciBI	YGGCCR	YGGCCR	
EciCI	CCTNAGG	CCTNAGG	

EciDI	CCSGG	CCSGG	
EciEI	GGGCCC	GGGCCC	
EclI	CAGCTG	CAGCTG	
EclII	CCWGG	CCWGG	
Ecl1I	CCGCGG	CCGCGG	
Ecl28I	CCGCGG	CCGCGG	
Ecl37I	CCGCGG	CCGCGG	
Ecl66I	CCWGG	CCWGG	
Ecl77I	CTGCAG	CTGCAG	
Ecl133I	CTGCAG	CTGCAG	
Ecl136I	CCWGG	CCWGG	
Ecl136II	GAGCTC	GAGCTC	F.
Ecl137I	GAGCTC	GAGCTC	
Ecl137II	CCWGG	CCWGG	
Ecl593I	CTGCAG	CTGCAG	
EclHKI	GACNNNNNGTC	GACNNNNNGTC	R.
EclJI	CGATCG	CGATCG	
EclRI	CCCGGG	CCCGGG	
EclS39I	CCWGG	CCWGG	
EclXI	CGGCCG	CGGCCG	MS.
Ecl18kI	CCNGG	CCNGG	
M.Ecl18kI	CCNGG	CCNGG	
Ecl37kI	CTGCAG	CTGCAG	
Ecl37kII	CCWGG	CCWGG	
Ecl54kI	CCWGG	CCWGG	
Ecl57kI	CCWGG	CCWGG	
Ecl699kI	CTGCAG	CTGCAG	
Ecl1zI	CTGCAG	CTGCAG	
Ecl1zII	CCWGG	CCWGG	
Ecl2zI	CTGCAG	CTGCAG	
Eco17I	GATATC	GATATC	
Eco24I	GRGCYC	GRGCYC	F.
Eco25I	GRGCYC	GRGCYC	
Eco26I	GRGCYC	GRGCYC	
Eco31I	GGTCTC	GAGACC	F.
M1.Eco31I	?	?	
M2.Eco31I	?	?	
Eco32I	GATATC	GATATC	F.
M.Eco32I	GATATC	GATATC	
Eco35I	GRGCYC	GRGCYC	
Eco37I	GGANNNNNNNNATGC	GCATNNNNNNNNTCC	
M.Eco37I	GGANNNNNNNNATGC	GGANNNNNNNNATGC	
Eco38I	CCWGG	CCWGG	
Eco39I	GGNCC	GGNCC	
Eco40I	CCWGG	CCWGG	
Eco41I	CCWGG	CCWGG	
Eco42I	GGTCTC	GAGACC	
Eco43I	CCNGG	CCNGG	
Eco47I	GGWCC	GGWCC	FO.
Eco47II	GGNCC	GGNCC	
M.Eco47II	GGNCC	GGNCC	
Eco47III	AGCGCT	AGCGCT	FGMOR.
M.Eco47III	AGCGCT	AGCGCT	
Eco48I	CTGCAG	CTGCAG	
Eco49I	CTGCAG	CTGCAG	
Eco50I	GGYRCC	GGYRCC	
Eco51I	GGTCTC	GAGACC	
Eco51II	CCNGG	CCNGG	
Eco52I	CGGCCG	CGGCCG	FKO.
Eco55I	CCGCGG	CCGCGG	
Eco56I	GCCGGC	GCCGGC	
M.Eco56I	GCCGGC	GCCGGC	
Eco57I	CTGAAG	CTTCAG	F.
M.Eco57I	CTGAAG	CTGAAG	
Eco60I	CCWGG	CCWGG	
Eco61I	CCWGG	CCWGG	
Eco64I	GGYRCC	GGYRCC	
M.Eco64I	GGYRCC	GGYRCC	
Eco65I	AAGCTT	AAGCTT	
Eco67I	CCWGG	CCWGG	
Eco68I	GRGCYC	GRGCYC	
Eco70I	CCWGG	CCWGG	
Eco71I	CCWGG	CCWGG	
Eco72I	CACGTG	CACGTG	F.
M.Eco72I	CACGTG	CACGTG	
Eco76I	CCTNAGG	CCTNAGG	
Eco78I	GGCGCC	GGCGCC	
Eco80I	CCNGG	CCNGG	
Eco81I	CCTNAGG	CCTNAGG	AFKO.
Eco82I	GAATTC	GAATTC	

Eco83I	CTGCAG	CTGCAG	
Eco85I	CCNGG	CCNGG	
Eco88I	CYCGRG	CYCGRG	F.
M.Eco88I	CYCGRG	CYCGRG	
Eco90I	YGGCCR	YGGCCR	
Eco91I	GGTNACC	GGTNACC	F.
Eco92I	CCGCGG	CCGCGG	
Eco93I	CCNGG	CCNGG	
Eco95I	GGTCTC	GAGACC	
Eco96I	CCGCGG	CCGCGG	
Eco97I	GGTCTC	GAGACC	
Eco98I	AAGCTT	AAGCTT	
M.Eco98I	AAGCTT	AAGCTT	
Eco99I	CCGCGG	CCGCGG	
Eco100I	CCGCGG	CCGCGG	
Eco101I	GGTCTC	GAGACC	
Eco104I	CCGCGG	CCGCGG	
Eco105I	TACGTA	TACGTA	FO.
M.Eco105I	TACGTA	TACGTA	
Eco112I	CTGAAG	CTTCAG	
Eco113I	GRGCYC	GRGCYC	
Eco115I	CCTNAGG	CCTNAGG	
Eco118I	CCTNAGG	CCTNAGG	
Eco120I	GGTCTC	GAGACC	
Eco121I	CCSGG	CCSGG	
Eco125I	CTGAAG	CTTCAG	
Eco127I	GGTCTC	GAGACC	
Eco128I	CCWGG	CCWGG	
M.Eco128I	CCWGG	CCWGG	
Eco129I	GGTCTC	GAGACC	
Eco130I	CCWWGG	CCWWGG	F.
Eco134I	CCGCGG	CCGCGG	
Eco135I	CCGCGG	CCGCGG	
Eco143I	GCGCGC	GCGCGC	
Eco147I	AGGCCT	AGGCCT	F.
M.Eco147I	AGGCCT	AGGCCT	
Eco149I	GGTACC	GGTACC	
Eco151I	CCGCGG	CCGCGG	
Eco152I	GCGCGC	GCGCGC	
Eco153I	CCNGG	CCNGG	
Eco155I	GGTCTC	GAGACC	
Eco156I	GGTCTC	GAGACC	
Eco157I	GGTCTC	GAGACC	
Eco158I	CCGCGG	CCGCGG	
Eco158II	TACGTA	TACGTA	
Eco159I	GAATTC	GAATTC	
Eco161I	CTGCAG	CTGCAG	
Eco162I	GGTCTC	GAGACC	
Eco164I	YGGCCR	YGGCCR	
Eco167I	CTGCAG	CTGCAG	
Eco168I	GGYRCC	GGYRCC	
Eco169I	GGYRCC	GGYRCC	
Eco170I	CCWGG	CCWGG	
Eco171I	GGYRCC	GGYRCC	
Eco173I	GGYRCC	GGYRCC	
Eco178I	GATATC	GATATC	
Eco179I	CCSGG	CCSGG	
Eco180I	GRGCYC	GRGCYC	
Eco182I	CCGCGG	CCGCGG	
Eco185I	GGTCTC	GAGACC	
Eco188I	AAGCTT	AAGCTT	
Eco190I	CCSGG	CCSGG	
Eco191I	GGTCTC	GAGACC	
Eco193I	CCWGG	CCWGG	
Eco195I	GGYRCC	GGYRCC	
Eco196I	CCGCGG	CCGCGG	
Eco196II	GGNCC	GGNCC	
Eco200I	CCNGG	CCNGG	
Eco201I	GGNCC	GGNCC	
Eco203I	GGTCTC	GAGACC	
Eco204I	GGTCTC	GAGACC	
Eco205I	GGTCTC	GAGACC	
Eco206I	CCWGG	CCWGG	
Eco207I	CCWGG	CCWGG	
Eco208I	CCGCGG	CCGCGG	
Eco208II	CCWWGG	CCWWGG	
Eco211I	GRGCYC	GRGCYC	
Eco215I	GRGCYC	GRGCYC	
Eco216I	GRGCYC	GRGCYC	
Eco217I	GGTCTC	GAGACC	

Eco225I	GGTCTC	GAGACC
Eco228I	GAATTC	GAATTC
Eco231I	AAGCTT	AAGCTT
M.Eco231I	AAGCTT	AAGCTT
Eco232I	GRGCYC	GRGCYC
Eco233I	GGTCTC	GAGACC
Eco237I	GAATTC	GAATTC
Eco239I	GGTCTC	GAGACC
Eco240I	GGTCTC	GAGACC
Eco241I	GGTCTC	GAGACC
Eco246I	GGTCTC	GAGACC
Eco247I	GGTCTC	GAGACC
Eco249I	GRGCYC	GRGCYC
Eco252I	GAATTC	GAATTC
Eco254I	CCWGG	CCWGG
Eco255I	AGTACT	AGTACT
M.Eco255I	AGTACT	AGTACT
Eco256I	CCWGG	CCWGG
Eco260I	CTGCAG	CTGCAG
Eco261I	CTGCAG	CTGCAG
Eco262I	GRGCYC	GRGCYC
Eco263I	GGTCTC	GAGACC
Eco377I	GGANNNNNNNNATGC	GCATNNNNNNNNTCC
M.Eco377I	GGANNNNNNNNATGC	GGANNNNNNNNATGC
Eco394I	GACNNNNNRTAAY	RTTAYNNNNNGTC
M.Eco394I	GACNNNNNRTAAY	GACNNNNNRTAAY
Eco585I	GCCNNNNNNTGCG	CGCANNNNNNGGC
M.Eco585I	GCCNNNNNNTGCG	GCCNNNNNNTGCG
Eco646I	CCANNNNNNNCTTC	GAAGNNNNNNNTGG
M.Eco646I	CCANNNNNNNCTTC	CCANNNNNNNCTTC
Eco777I	GGANNNNNNTATC	GATANNNNNNTCC
M.Eco777I	GGANNNNNNTATC	GGANNNNNNTATC
Eco826I	GCANNNNNNCTGA	TCAGNNNNNNTGC
M.Eco826I	GCANNNNNNCTGA	GCANNNNNNCTGA
Eco851I	GTCANNNNNNTGAY	RTCANNNNNNTGAC
M.Eco851I	GTCANNNNNNTGAY	GTCANNNNNNTGAY
Eco912I	CACNNNNNTGGC	GCCANNNNNGTG
M.Eco912I	CACNNNNNTGGC	CACNNNNNTGGC
Eco1158I	TGANNNNNNNNTGCT	AGCANNNNNNNNTCA
M.Eco1158I	TGANNNNNNNNTGCT	TGANNNNNNNNTGCT
Eco1265I	TGANNNNNNNNTGCT	AGCANNNNNNNNTCA
M.Eco1265I	TGANNNNNNNNTGCT	TGANNNNNNNNTGCT
Eco1323I	GGANNNNNNNNATGC	GCATNNNNNNNNTCC
Eco1341I	CCANNNNNNNCTTC	GAAGNNNNNNNTGG
Eco1342I	AACNNNNNNGTGC	GCACNNNNNNGTT
Eco1344I	AACNNNNNNGTGC	GCACNNNNNNGTT
Eco1344II	GGANNNNNNNNATGC	GCATNNNNNNNNTCC
Eco1348I	GGANNNNNNTATC	GATANNNNNNTCC
Eco1383I	CCANNNNNNNCTTC	GAAGNNNNNNNTGG
Eco1386I	GGANNNNNNNNATGC	GCATNNNNNNNNTCC
Eco1394I	AACNNNNNNGTGC	GCACNNNNNNGTT
Eco1412I	GGANNNNNNTATC	GATANNNNNNTCC
Eco1413I	CCANNNNNNNCTTC	GAAGNNNNNNNTGG
Eco1422I	CCANNNNNNNCTTC	GAAGNNNNNNNTGG
Eco1424I	CCANNNNNNNCTTC	GAAGNNNNNNNTGG
Eco1427I	GGANNNNNNNNATGC	GCATNNNNNNNNTCC
Eco1430I	GGANNNNNNNNATGC	GCATNNNNNNNNTCC
Eco1432I	CCANNNNNNNCTTC	GAAGNNNNNNNTGG
Eco1441I	TGANNNNNNNNTGCT	AGCANNNNNNNNTCA
Eco1443I	TGANNNNNNNNTGCT	AGCANNNNNNNNTCA
Eco1446I	GAGNNNNNNNGTCA	TGACNNNNNNNCTC
Eco1447I	TGANNNNNNNNTGCT	AGCANNNNNNNNTCA
Eco1455I	GCANNNNNNCTGA	TCAGNNNNNNTGC
Eco1456I	GGANNNNNNNNATGC	GCATNNNNNNNNTCC
Eco1476I	GGANNNNNNNNATGC	GCATNNNNNNNNTCC
Eco1524I	AGGCCT	AGGCCT
Eco1831I	CCSGG	CCSGG
M.Eco1831I	CCSGG	CCSGG
Eco14444I	TGANNNNNNNNTGCT	AGCANNNNNNNNTCA
EcoAI	GAGNNNNNNNGTCA	TGACNNNNNNNCTC
M.EcoAI	GAGNNNNNNNGTCA	GAGNNNNNNNGTCA
EcoA4I	GGTCTC	GAGACC
EcoBI	TGANNNNNNNNTGCT	AGCANNNNNNNNTCA
M.EcoBI	TGANNNNNNNNTGCT	TGANNNNNNNNTGCT
EcoCKI	?	?
EcoDI	TTANNNNNNNGTCY	RGACNNNNNNNTAA
M.EcoDI	TTANNNNNNNGTCY	TTANNNNNNNGTCY
EcoDR2	TCANNNNNNGTCG	CGACNNNNNNTGA
M.EcoDR2	TCANNNNNNGTCG	TCANNNNNNGTCG
EcoDR3	TCANNNNNNNATCG	CGATNNNNNNNTGA

M.EcoDR3	TCANNNNNNNATCG	TCANNNNNNNATCG	
EcoDXXI	TCANNNNNNNRTTC	GAAYNNNNNNNTGA	
M.EcoDXXI	TCANNNNNNNRTTC	TCANNNNNNNRTTC	
M.Eco67Dam	GATC	GATC	
EcoEI	GAGNNNNNNNATGC	GCATNNNNNNNCTC	
M.EcoEI	GAGNNNNNNNATGC	GAGNNNNNNNATGC	
EcoHI	CCSGG	CCSGG	
M.EcoHI	CCSGG	CCSGG	
EcoHAI	YGGCCR	YGGCCR	
EcoHK31I	YGGCCR	YGGCCR	
M.EcoHK31I	YGGCCR	YGGCCR	
EcoICRI	GAGCTC	GAGCTC	IRV.
EcoKI	AACNNNNNNGTGC	GCACNNNNNNGTT	
M.EcoKI	AACNNNNNNGTGC	AACNNNNNNGTGC	
Eco71KI	GGTCTC	GAGACC	
Eco75KI	GRGCYC	GRGCYC	
M.EcoKDam	GATC	GATC	N.
M.EcoKDcm	CCWGG	CCWGG	
Eco57MI	CTGRAG	CTYCAG	F.
EcoNI	CCTNNNNNAGG	CCTNNNNNAGG	N.
M.EcoNI	CCTNNNNNAGG	CCTNNNNNAGG	
EcoO34I	?	?	
EcoO44I	GGTCTC	GAGACC	
EcoO65I	GGTNACC	GGTNACC	K.
EcoO109I	RGGNCCY	RGGNCCY	AFJKN.
M.EcoO109I	RGGNCCY	RGGNCCY	
EcoO128I	GGTNACC	GGTNACC	
EcoPI	AGACC	GGTCT	
M.EcoPI	AGACC	AGACC	
EcoP15I	CAGCAG	CTGCTG	N.
M.EcoP15I	CAGCAG	CAGCAG	
M.EcoP1Dam	GATC	GATC	
EcoRI	GAATTC	GAATTC	ABCFGHIJKMNOQRSUVXY.
M.EcoRI	GAATTC	GAATTC	JKN.
EcoRII	CCWGG	CCWGG	FJMOS.
M.EcoRII	CCWGG	CCWGG	
EcoRV	GATATC	GATATC	ABCGHIJKMNOQRSUVXY.
M.EcoRV	GATATC	GATATC	
EcoR5I	?	?	
M.EcoR5I	?	?	
EcoR9I	?	?	
M.EcoR9I	?	?	
EcoR10I	?	?	
M.EcoR10I	?	?	
EcoR11I	?	?	
M.EcoR11I	?	?	
EcoR12I	?	?	
M.EcoR12I	?	?	
EcoR13I	?	?	
M.EcoR13I	?	?	
EcoR15I	?	?	
M.EcoR15I	?	?	
EcoR17I	?	?	
M.EcoR17I	?	?	
EcoR23I	?	?	
M.EcoR23I	?	?	
EcoR24I	?	?	
M.EcoR24I	?	?	
EcoR25I	?	?	
M.EcoR25I	?	?	
EcoR42I	?	?	
M.EcoR42I	?	?	
EcoR70I	?	?	
M.EcoR70I	?	?	
EcoR124I	GAANNNNNNRTCG	CGAYNNNNNNTTC	
M.EcoR124I	GAANNNNNNRTCG	GAANNNNNNRTCG	
EcoR124II	GAANNNNNNNRTCG	CGAYNNNNNNNTTC	
M.EcoR124II	GAANNNNNNNRTCG	GAANNNNNNNRTCG	
EcoRD2	GAANNNNNNRTTC	GAAYNNNNNNTTC	
M.EcoRD2	GAANNNNNNRTTC	GAANNNNNNRTTC	
EcoRD3	GAANNNNNNNRTTC	GAAYNNNNNNNTTC	
M.EcoRD3	GAANNNNNNNRTTC	GAANNNNNNNRTTC	
F-EcoT5I	TGGCGACGAAAACCGCTTGGAAAGTGGCTG	CAGCCACTTTCCAAGCGGTTTTCGTCGCCA	
F-EcoT5II	ACCTACCATTAACGGAGTCAAAGGCCATTG	CAATGGCCTTTGACTCCGTTAATGGTAGGT	
F-EcoT5IV	TAGGTACTGGACTTAAAATTCAGGTTTTGT	ACAAAACCTGAATTTTAAGTCCAGTACCTA	
EcoT14I	CCWWGG	CCWWGG	K.
EcoT22I	ATGCAT	ATGCAT	AKO.
M.EcoT22I	ATGCAT	ATGCAT	
EcoT38I	GRGCYC	GRGCYC	J.
M.EcoT38I	GRGCYC	GRGCYC	

EcoT88I	GRGCYC	GRGCYC	
EcoT93I	GRGCYC	GRGCYC	
EcoT95I	GRGCYC	GRGCYC	
EcoT104I	CCWWGG	CCWWGG	
M.EcoT1Dam	GATC	GATC	
M.EcoT2Dam	GATC	GATC	
M.EcoT4Dam	GATC	GATC	
EcoVIII	AAGCTT	AAGCTT	
M.EcoVIII	AAGCTT	AAGCTT	
M.EcoVT2Dam	GATC	GATC	
Eco13kI	CCNGG	CCNGG	
Eco21kI	CCNGG	CCNGG	
Eco27kI	CYCGRG	CYCGRG	
Eco29kI	CCGCGG	CCGCGG	
M.Eco29kI	CCGCGG	CCGCGG	
Eco53kI	GAGCTC	GAGCTC	
Eco110kI	CCTNAGG	CCTNAGG	
Eco137kI	CCNGG	CCNGG	
EcoprrI	CCANNNNNNNRTGC	GCAYNNNNNNNTGG	
M.EcoprrI	CCANNNNNNNRTGC	CCANNNNNNNRTGC	
M.EfaBMDam	GATC	GATC	
EgeI	GGCGCC	GGCGCC	I.
EheI	GGCGCC	GGCGCC	FO.
ErhI	CCWWGG	CCWWGG	IV.
ErhB9I	CGATCG	CGATCG	
ErhB9II	CCWWGG	CCWWGG	
ErpI	GGWCC	GGWCC	
M.EsaBC1I	AGCT	AGCT	
M.EsaBC2I	?	?	
EsaBC3I	TCGA	TCGA	
M.EsaBC3I	TCGA	TCGA	
EsaBC4I	GGCC	GGCC	
M.EsaBC4I	GGCC	GGCC	
M.EsaBS1I	CATG	CATG	
M.EsaBS2I	?	?	
EsaBS9I	CGCG	CGCG	
M.EsaBS9I	CGCG	CGCG	
M.EsaDix1I	TTTAAA	TTTAAA	
M.EsaDix2I	TCGA	TCGA	
M.EsaDix3I	TCGA	TCGA	
M.EsaDix4I	TTAA	TTAA	
M.EsaDix5I	TTAA	TTAA	
M.EsaDix6I	TCGA	TCGA	
M.EsaDix7I	GGCC	GGCC	
EsaLHCI	GATC	GATC	
M.EsaLHCI	GATC	GATC	
M.EsaLHCII	?	?	
M.EsaLHCIII	GATC	GATC	
M.EsaLHC2I	?	?	
M1.EsaS1I	GGCC	GGCC	
M2.EsaS1I	GGCC	GGCC	
M.EsaS3I	GATC	GATC	
M.EsaS4I	AGCT	AGCT	
M.EsaS5I	?	?	
M.EsaS6I	CTAG	CTAG	
M.EsaS7I	CTAG	CTAG	
M.EsaS8I	GATC	GATC	
M.EsaS9I	?	?	
M.EsaWC1I	GGCC	GGCC	
M.EsaWC2I	GANTC	GANTC	
M.EsaWC2II	CCTNAGG	CCTNAGG	
M.EsaWC3I	TCGA	TCGA	
M.EsaWC4I	TCGA	TCGA	
EscI	CTCGAG	CTCGAG	
Ese3I	CCGCGG	CCGCGG	
Ese4I	GRGCYC	GRGCYC	
Ese6I	CCGCGG	CCGCGG	
Ese6II	CCWGG	CCWGG	
EspI	GCTNAGC	GCTNAGC	
EspII	?	?	
Esp1I	GGYRCC	GGYRCC	
Esp2I	CCWGG	CCWGG	
Esp3I	CGTCTC	GAGACG	F.
M.Esp3I	CGTCTC	CGTCTC	
Esp4I	CTTAAG	CTTAAG	
Esp5I	GCCGGC	GCCGGC	
Esp5II	CTGCAG	CTGCAG	
Esp6I	GGYRCC	GGYRCC	
Esp7I	GCGCGC	GCGCGC	
Esp8I	GCGCGC	GCGCGC	

Esp9I	GGYRCC	GGYRCC	
Esp10I	GGYRCC	GGYRCC	
Esp11I	GGYRCC	GGYRCC	
Esp12I	GGYRCC	GGYRCC	
Esp13I	GGYRCC	GGYRCC	
Esp14I	GGYRCC	GGYRCC	
Esp15I	GGYRCC	GGYRCC	
Esp16I	CGTCTC	GAGACG	
Esp19I	GGTACC	GGTACC	
Esp21I	GGYRCC	GGYRCC	
Esp22I	GGYRCC	GGYRCC	
Esp23I	CGTCTC	GAGACG	
Esp24I	CCWGG	CCWGG	
Esp25I	GGYRCC	GGYRCC	
Esp141I	CTGCAG	CTGCAG	
Esp1396I	CCANNNNNTGG	CCANNNNNTGG	
M.Esp1396I	CCANNNNNTGG	CCANNNNNTGG	
EspHK7I	CCWGG	CCWGG	
EspHK16I	YGGCCR	YGGCCR	
EspHK22I	CCWGG	CCWGG	
EspHK24I	YGGCCR	YGGCCR	
EspHK26I	TCCGGA	TCCGGA	
EspHK29I	CYCGRG	CYCGRG	
EspHK30I	CCWGG	CCWGG	
FaeI	CATG	CATG	I.
FalI	AAGNNNNNCTT	AAGNNNNNCTT	I.
FalI	AAGNNNNNCTT	AAGNNNNNCTT	I.
FalII	CGCG	CGCG	
FaqI	GGGAC	GTCCC	F.
FatI	CATG	CATG	IN.
FauI	CCCGC	GCGGG	IN.
M1.FauI	CCCGC	CCCGC	
FauBI	?	?	
FauBII	CGCG	CGCG	
FauNDI	CATATG	CATATG	IV.
FbaI	TGATCA	TGATCA	AK.
FblI	GTMKAC	GTMKAC	IV.
FbrI	GCNGC	GCNGC	
FdiI	GGWCC	GGWCC	
FdiII	TGCGCA	TGCGCA	
FgoI	CTAG	CTAG	
FinI	GGGAC	GTCCC	
FinII	CCGG	CCGG	
FinSI	GGCC	GGCC	
FisI	CTAG	CTAG	
FmuI	GGNCC	GGNCC	
Fnu48I	?	?	
FnuAI	GANTC	GANTC	
FnuAII	GATC	GATC	
FnuCI	GATC	GATC	
FnuDI	GGCC	GGCC	
M.FnuDI	GGCC	GGCC	
FnuDII	CGCG	CGCG	
M.FnuDII	CGCG	CGCG	
FnuDIII	GCGC	GCGC	
M.FnuDIII	GCGC	GCGC	
FnuEI	GATC	GATC	
Fnu4HI	GCNGC	GCNGC	N.
M.Fnu4HI	GCNGC	GCNGC	
FokI	GGATG	CATCC	AGIJKMNRV.
M.FokI	GGATG	GGATG	
FriOI	GRGCYC	GRGCYC	IV.
FscI	CCGCGG	CCGCGG	
FseI	GGCCGGCC	GGCCGGCC	AKN.
M.FseI	GGCCGGCC	GGCCGGCC	
FsfI	CTGAAG	CTTCAG	
FsiI	RAATTY	RAATTY	
FspI	TGCGCA	TGCGCA	JNO.
M.FspI	TGCGCA	TGCGCA	
FspII	TTCGAA	TTCGAA	
Fsp1604I	CCWGG	CCWGG	
FspAI	RTGCGCAY	RTGCGCAY	F.
FspBI	CTAG	CTAG	F.
Fsp4HI	GCNGC	GCNGC	I.
M.Fsp4HI	GCNGC	GCNGC	I.
FspMI	CGCG	CGCG	
FspMSI	GGWCC	GGWCC	
FssI	GGWCC	GGWCC	
M.FssI	GGWCC	GGWCC	
FsuI	GACNNNGTC	GACNNNGTC	

FunI	AGCGCT	AGCGCT	
FunII	GAATTC	GAATTC	
M.Fvi3I	?	?	
GalI	CCGCGG	CCGCGG	
GceI	CCGCGG	CCGCGG	
GceGLI	CCGCGG	CCGCGG	
GdiI	AGGCCT	AGGCCT	
GdiII	CGGCCR	YGGCCG	
GdoI	GGATCC	GGATCC	
M.GgaDnmt1	?	?	
GglI	?	?	
GinI	GGATCC	GGATCC	
GobAI	AGGCCT	AGGCCT	
GoxI	GGATCC	GGATCC	
GseI	GGNCC	GGNCC	
GseII	CTGCAG	CTGCAG	
GseIII	GGATCC	GGATCC	
GspI	CAGCTG	CAGCTG	
GspAI	GGWCC	GGWCC	
GspAII	TGCGCA	TGCGCA	
GspAIII	?	?	
GstI	GGATCC	GGATCC	
Gst1588I	CYCGRG	CYCGRG	
Gst1588II	GATC	GATC	
GsuI	CTGGAG	CTCCAG	F.
M.GsuI	CTGGAG	CTGGAG	
M.H2I	GGCC	GGCC	
HacI	GATC	GATC	
HaeI	WGGCCW	WGGCCW	
HaeII	RGCGCY	RGCGCY	GJKMNORS.
M.HaeII	RGCGCY	RGCGCY	
HaeIII	GGCC	GGCC	ABGHIJKMNOQRSUXY.
M.HaeIII	GGCC	GGCC	KN.
HaeIV	GAYNNNNNRTC	GAYNNNNNRTC	
HaeIV	GAYNNNNNRTC	GAYNNNNNRTC	
HagI	?	?	
HalI	GAATTC	GAATTC	
HalII	CTGCAG	CTGCAG	
Hal22I	GAATTC	GAATTC	
HapI	?	?	
HapII	CCGG	CCGG	AK.
M.HapII	CCGG	CCGG	K.
HcuI	?	?	
HgaI	GACGC	GCGTC	IN.
M1.HgaI	GACGC	GACGC	
M2.HgaI	GACGC	GACGC	
HgiI	GRCGYC	GRCGYC	
HgiAI	GWGCWC	GWGCWC	
M.HgiAI	GWGCWC	GWGCWC	
HgiBI	GGWCC	GGWCC	
M.HgiBI	GGWCC	GGWCC	
HgiCI	GGYRCC	GGYRCC	
M.HgiCI	GGYRCC	GGYRCC	
HgiCII	GGWCC	GGWCC	
M.HgiCII	GGWCC	GGWCC	
HgiCIII	GTCGAC	GTCGAC	
HgiDI	GRCGYC	GRCGYC	
M.HgiDI	GRCGYC	GRCGYC	
HgiDII	GTCGAC	GTCGAC	
M.HgiDII	GTCGAC	GTCGAC	
HgiEI	GGWCC	GGWCC	
M.HgiEI	GGWCC	GGWCC	
HgiEII	ACCNNNNNNGGT	ACCNNNNNNGGT	
HgiFI	?	?	
HgiGI	GRCGYC	GRCGYC	
M.HgiGI	GRCGYC	GRCGYC	
HgiHI	GGYRCC	GGYRCC	
HgiHII	GRCGYC	GRCGYC	
HgiHIII	GGWCC	GGWCC	
HgiJI	GGWCC	GGWCC	
HgiJII	GRGCYC	GRGCYC	
HgiKI	?	?	
HgiS21I	CCSGG	CCSGG	
HgiS22I	CCSGG	CCSGG	
HhaI	GCGC	GCGC	ABFGJKNORUY.
M.HhaI	GCGC	GCGC	N.
HhaII	GANTC	GANTC	
M.HhaII	GANTC	GANTC	
HhdI	CCWGG	CCWGG	
HhgI	GGCC	GGCC	

HhlI	?	?	
Hin1I	GRCGYC	GRCGYC	FKO.
Hin1II	CATG	CATG	F.
M.Hin1II	CATG	CATG	
Hin2I	CCGG	CCGG	
Hin3I	CCSGG	CCSGG	
Hin4I	GAYNNNNNVTC	GABNNNNNRTC	F.
Hin4I	GABNNNNNRTC	GAYNNNNNVTC	F.
Hin4II	CCTTC	GAAGG	
Hin5I	CCGG	CCGG	
Hin5II	GGNCC	GGNCC	
Hin5III	AAGCTT	AAGCTT	
Hin6I	GCGC	GCGC	F.
Hin7I	GCGC	GCGC	
Hin8I	GRCGYC	GRCGYC	
Hin8II	CATG	CATG	
Hin173I	AAGCTT	AAGCTT	
Hin1056I	CGCG	CGCG	
Hin1056II	?	?	
Hin1076III	AAGCTT	AAGCTT	
Hin1160II	GTYRAC	GTYRAC	
Hin1161II	GTYRAC	GTYRAC	
HinGUI	GCGC	GCGC	
HinGUII	GGATG	CATCC	
HinHI	RGCGCY	RGCGCY	
M.HinHP1Dam	GATC	GATC	
M.HinHP2Dam	GATC	GATC	
HinJCI	GTYRAC	GTYRAC	
HinJCII	AAGCTT	AAGCTT	
HinP1I	GCGC	GCGC	N.
M.HinP1I	GCGC	GCGC	
HinS1I	GCGC	GCGC	
HinS2I	GCGC	GCGC	
HinSAFI	AAGCTT	AAGCTT	
HinbIII	AAGCTT	AAGCTT	
HincII	GTYRAC	GTYRAC	ABFGHJKNOQRUXY.
M.HincII	GTYRAC	GTYRAC	
HindI	CAC	GTG	
M.HindI	CAC	CAC	
HindII	GTYRAC	GTYRAC	IMSV.
M.HindII	GTYRAC	GTYRAC	
HindIII	AAGCTT	AAGCTT	ABCFGHIJKMNOQRSUVXY.
M.HindIII	AAGCTT	AAGCTT	K.
M.HindV	GRCGYC	GRCGYC	
M.HindDam	GATC	GATC	
HineI	CGAAT	ATTCG	
HinfI	GANTC	GANTC	ABCFGHIJKMNOQRUVXY.
M.HinfI	GANTC	GANTC	
HinfII	AAGCTT	AAGCTT	
HinfIII	CGAAT	ATTCG	
M.HinfIII	CGAAT	CGAAT	
HjaI	GATATC	GATATC	
M.HjaI	GATATC	GATATC	
I-HmuI	AGTAATGAGCCTAACGCTCAGCAA	TTGCTGAGCGTTAGGCTCATTACT	
I-HmuII	AGTAATGAGCCTAACGCTCAACAA	TTGTTGAGCGTTAGGCTCATTACT	
HpaI	GTTAAC	GTTAAC	ABCGHIJKMNOQRSUVX.
M.HpaI	GTTAAC	GTTAAC	
HpaII	CCGG	CCGG	BFGIMNOQRSUVX.
M.HpaII	CCGG	CCGG	N.
HphI	GGTGA	TCACC	FN.
M1.HphI	GGTGA	GGTGA	
M2.HphI	GGTGA	GGTGA	
M.HpyI	CATG	CATG	
HpyII	GAAGA	TCTTC	
M.HpyIII	?	?	
HpyIV	GANTC	GANTC	
HpyV	TCGA	TCGA	
HpyVIII	CCGG	CCGG	
Hpy8I	GTNNAC	GTNNAC	F.
M.Hpy8I	GTNNAC	GTNNAC	
Hpy8II	GTSAC	GTSAC	
Hpy8III	GWGCWC	GWGCWC	
Hpy26I	TGCA	TGCA	
Hpy26II	TCGA	TCGA	
M.Hpy26III	?	?	
Hpy51I	GTSAC	GTSAC	
Hpy99I	CGWCG	CGWCG	N.
M.Hpy99I	CGWCG	CGWCG	
Hpy99II	GTSAC	GTSAC	
M.Hpy99II	GTSAC	GTSAC	

Hpy99III	GCGC	GCGC	
M.Hpy99III	GCGC	GCGC	
Hpy99IV	CCNNGG	CCNNGG	
M.Hpy99IV	CCNNGG	CCNNGG	
M1.Hpy99V	CCTC	CCTC	
M.Hpy99VI	GATC	GATC	
M.Hpy99VII	?	?	
M.Hpy99VIII	CCGG	CCGG	
M.Hpy99IX	GANTC	GANTC	
M.Hpy99X	CATG	CATG	
M.Hpy99XI	ACGT	ACGT	
Hpy166I	TCNGA	TCNGA	
Hpy166II	GTNNAC	GTNNAC	
Hpy166III	CCTC	GAGG	
M.Hpy166IV	CATG	CATG	
Hpy178II	GAAGA	TCTTC	
Hpy178III	TCNNGA	TCNNGA	
Hpy178VI	GGATG	CATCC	
Hpy178VII	GGCC	GGCC	
Hpy188I	TCNGA	TCNGA	N.
M.Hpy188I	TCNGA	TCNGA	
M.Hpy188II	CATG	CATG	
Hpy188III	TCNNGA	TCNNGA	N.
M.Hpy188III	TCNNGA	TCNNGA	
M.Hpy788180	?	?	
M.HpyAI	CATG	CATG	
HpyAII	GAAGA	TCTTC	
M1.HpyAII	GAAGA	GAAGA	
M2.HpyAII	GAAGA	GAAGA	
HpyAIII	GATC	GATC	
M.HpyAIII	GATC	GATC	
HpyAIV	GANTC	GANTC	
M.HpyAIV	GANTC	GANTC	
HpyAV	CCTTC	GAAGG	
M.HpyAV	CCTTC	CCTTC	
M1.HpyAVI	CCTC	CCTC	
M2.HpyAVI	CCTC	CCTC	
M.HpyAVII	ATTAAT	ATTAAT	
M.HpyAVIII	GCGC	GCGC	
M.HpyAIX	GTNNAC	GTNNAC	
M.HpyAX	TCGA	TCGA	
M.HpyAXI	?	?	
Hpy87AI	GANTC	GANTC	
M.Hpy87AI	GANTC	GANTC	
HpyBI	GTAC	GTAC	
HpyBII	GTNNAC	GTNNAC	
HpyCI	GATATC	GATATC	
HpyCII	?	?	
HpyC1I	CCATC	GATGG	
M1.HpyC1I	CCATC	CCATC	
M2.HpyC1I	CCATC	CCATC	
HpyCH4I	CATG	CATG	
HpyCH4II	CTNAG	CTNAG	
HpyCH4III	ACNGT	ACNGT	N.
HpyCH4IV	ACGT	ACGT	N.
M.HpyCH4IV	ACGT	ACGT	
HpyCH4V	TGCA	TGCA	N.
M.HpyCH4V	TGCA	TGCA	
HpyCH4VI	TCNNGA	TCNNGA	
HpyF1I	GTSAC	GTSAC	
HpyF2I	CTRYAG	CTRYAG	
HpyF2II	GANTC	GANTC	
HpyF3I	CTNAG	CTNAG	F.
HpyF4I	GTSAC	GTSAC	
HpyF4II	CTNAG	CTNAG	
HpyF5I	CTNAG	CTNAG	
HpyF5II	ACNGT	ACNGT	
HpyF6I	GGATG	CATCC	
HpyF6II	GTSAC	GTSAC	
HpyF6III	CTNAG	CTNAG	
HpyF7I	CTNAG	CTNAG	
HpyF7II	GWGCWC	GWGCWC	
HpyF7III	GTNNAC	GTNNAC	
HpyF9I	GTSAC	GTSAC	
HpyF9II	CTNAG	CTNAG	
HpyF9III	ACNGT	ACNGT	
HpyF10I	GCGC	GCGC	
HpyF10II	GANTC	GANTC	
HpyF10III	CCNNGG	CCNNGG	
HpyF10IV	GTAC	GTAC	

HpyF10V	GGCC	GGCC	
HpyF10VI	GCNNNNNNNGC	GCNNNNNNNGC	F.
HpyF11I	CTNAG	CTNAG	
HpyF11II	TCNGA	TCNGA	
HpyF12I	ACNGT	ACNGT	
HpyF12II	TCNGA	TCNGA	
HpyF13I	GTSAC	GTSAC	
HpyF13II	CTNAG	CTNAG	
HpyF13III	ACGT	ACGT	
HpyF13IV	GTAC	GTAC	
HpyF14I	CGCG	CGCG	
HpyF14II	GTNNAC	GTNNAC	
HpyF14III	TCGA	TCGA	
HpyF15I	CGCG	CGCG	
HpyF15II	TCNGA	TCNGA	
HpyF16I	TCGA	TCGA	
HpyF16II	TCNNGA	TCNNGA	
HpyF17I	TCNGA	TCNGA	
M.HpyF17I	TCNGA	TCNGA	
HpyF18I	GANTC	GANTC	
HpyF19I	CTNAG	CTNAG	
HpyF19II	TCNGA	TCNGA	
HpyF19III	TCNNGA	TCNNGA	
HpyF20I	ACNGT	ACNGT	
HpyF21I	CTNAG	CTNAG	
HpyF21II	GTAC	GTAC	
HpyF22I	ACNGT	ACNGT	
HpyF22II	CTNAG	CTNAG	
HpyF22III	TCNNGA	TCNNGA	
HpyF23I	TCGA	TCGA	
HpyF24I	TCGA	TCGA	
HpyF24II	CTNAG	CTNAG	
HpyF25I	CTNAG	CTNAG	
HpyF25II	GTSAC	GTSAC	
HpyF26I	CGCG	CGCG	
HpyF26II	GGCC	GGCC	
HpyF26III	TCGA	TCGA	
HpyF27I	CTNAG	CTNAG	
HpyF27II	TCNGA	TCNGA	
HpyF28I	TCNGA	TCNGA	
HpyF29I	GGCC	GGCC	
HpyF30I	TCGA	TCGA	
HpyF30II	CTNAG	CTNAG	
HpyF31I	GTAC	GTAC	
HpyF31II	GTSAC	GTSAC	
HpyF32I	CTNAG	CTNAG	
HpyF33I	TCNGA	TCNGA	
HpyF33II	GGCC	GGCC	
HpyF34I	CTNAG	CTNAG	
HpyF34II	GTSAC	GTSAC	
HpyF35I	TCGA	TCGA	
HpyF35II	ACGT	ACGT	
HpyF35III	ACNGT	ACNGT	
HpyF35IV	GTSAC	GTSAC	
HpyF36I	GTSAC	GTSAC	
HpyF36II	GTAC	GTAC	
HpyF36III	TGCA	TGCA	
HpyF36IV	GDGCHC	GDGCHC	
HpyF37I	CTNAG	CTNAG	
HpyF38I	GANTC	GANTC	
HpyF38II	TGCA	TGCA	
HpyF40I	ACNGT	ACNGT	
HpyF40II	TCGA	TCGA	
HpyF40III	GTSAC	GTSAC	
HpyF41I	ACNGT	ACNGT	
HpyF41II	CTNAG	CTNAG	
HpyF42I	GGCC	GGCC	
HpyF42II	ACNGT	ACNGT	
HpyF42III	TCNGA	TCNGA	
HpyF42IV	TCGA	TCGA	
HpyF43I	CCGG	CCGG	
HpyF44I	GANTC	GANTC	
HpyF44II	GGNNCC	GGNNCC	
HpyF44III	TGCA	TGCA	
HpyF44IV	TCNNGA	TCNNGA	
HpyF44V	GTAC	GTAC	
HpyF45I	TCGA	TCGA	
HpyF45II	TGCA	TGCA	
HpyF46I	ACNGT	ACNGT	
HpyF46II	GWGCWC	GWGCWC	

HpyF46III	GTNNAC	GTNNAC	
HpyF46IV	TCNGA	TCNGA	
HpyF46V	GGCC	GGCC	
HpyF47I	GDGCHC	GDGCHC	
HpyF48I	GTSAC	GTSAC	
HpyF48II	ACNGT	ACNGT	
HpyF48III	TGCA	TGCA	
HpyF49I	TCGA	TCGA	
HpyF49II	GTSAC	GTSAC	
HpyF49III	GTNNAC	GTNNAC	
HpyF49IV	GGCC	GGCC	
HpyF49V	TGCA	TGCA	
HpyF50I	GTNNAC	GTNNAC	
HpyF50II	TCNGA	TCNGA	
HpyF51I	GTSAC	GTSAC	
HpyF51II	ACNGT	ACNGT	
HpyF52I	TCGA	TCGA	
HpyF52II	CGCG	CGCG	
HpyF52III	GTAC	GTAC	
HpyF53I	GGCC	GGCC	
HpyF53II	GTAC	GTAC	
HpyF54I	ACNGT	ACNGT	
HpyF55I	ACNGT	ACNGT	
HpyF55II	GANTC	GANTC	
HpyF56I	ACNGT	ACNGT	
HpyF57I	GGCC	GGCC	
HpyF58I	ACNGT	ACNGT	
HpyF59I	CTNAG	CTNAG	
HpyF59II	GTAC	GTAC	
HpyF59III	TCGA	TCGA	
HpyF60I	GANTC	GANTC	
HpyF60II	CTNAG	CTNAG	
HpyF61I	TCNGA	TCNGA	
HpyF61II	CCNNGG	CCNNGG	
HpyF61III	CGWCG	CGWCG	
HpyF62I	ACNGT	ACNGT	
HpyF62II	TCGA	TCGA	
HpyF62III	GTSAC	GTSAC	
HpyF63I	GGCC	GGCC	
HpyF64I	TCGA	TCGA	
HpyF64II	ACNGT	ACNGT	
HpyF64III	TCNGA	TCNGA	
HpyF64IV	CGCG	CGCG	
HpyF64V	CTNAG	CTNAG	
HpyF65I	ACNGT	ACNGT	
HpyF65II	TCGA	TCGA	
HpyF65III	GTAC	GTAC	
HpyF66I	GGNCC	GGNCC	
HpyF66II	CTNAG	CTNAG	
HpyF66III	GTAC	GTAC	
HpyF66IV	TCGA	TCGA	
HpyF67I	CTNAG	CTNAG	
HpyF67II	TGCA	TGCA	
HpyF67III	GGATG	CATCC	
HpyF67IV	CCNNGG	CCNNGG	
HpyF68I	ACNGT	ACNGT	
HpyF68II	CTNAG	CTNAG	
HpyF69I	ACNGT	ACNGT	
HpyF69II	GGCC	GGCC	
HpyF70I	CTNAG	CTNAG	
HpyF71I	TCGA	TCGA	
HpyF71II	GGNCC	GGNCC	
HpyF71III	GANTC	GANTC	
HpyF72I	GGCC	GGCC	
HpyF72II	CTNAG	CTNAG	
HpyF72III	GANTC	GANTC	
HpyF73I	GGNNCC	GGNNCC	
HpyF73II	TCGA	TCGA	
HpyF73III	GGCC	GGCC	
HpyF73IV	GGNCC	GGNCC	
HpyF74I	ACNGT	ACNGT	
HpyF74II	ACGT	ACGT	
HpyHPK5I	CTNAG	CTNAG	
HpyHPK5II	GATC	GATC	
HpyJP26I	TGCA	TGCA	
HpyJP26II	TCGA	TCGA	
HpyNI	CCNGG	CCNGG	
M.HsaDnmt1A	?	?	N.
M.HsaDnmt1B	?	?	
M.HsaDnmt3A	?	?	

M.HsaDnmt3B	?	?	
M.HsaDnmt3L	?	?	
HsoI	GCGC	GCGC	
Hsp2I	GGWCC	GGWCC	
Hsp92I	GRCGYC	GRCGYC	R.
Hsp92II	CATG	CATG	R.
HspAI	GCGC	GCGC	IV.
M.HspAI	GCGC	GCGC	
HsuI	AAGCTT	AAGCTT	
ItaI	GCNGC	GCNGC	M.
KasI	GGCGCC	GGCGCC	N.
M.KasI	GGCGCC	GGCGCC	
Kaz48kI	RGGNCCY	RGGNCCY	
KoxI	GGTNACC	GGTNACC	
KoxII	GRGCYC	GRGCYC	
Kox165I	CCWGG	CCWGG	
KoyI	GTCGAC	GTCGAC	
Kpl79I	CGATCG	CGATCG	
KpnI	GGTACC	GGTACC	ABCFGHIJKMNOQRSUVXY.
M.KpnI	GGTACC	GGTACC	
Kpn2I	TCCGGA	TCCGGA	F.
M.Kpn2I	TCCGGA	TCCGGA	
Kpn10I	CCWGG	CCWGG	
Kpn12I	CTGCAG	CTGCAG	
Kpn13I	CCWGG	CCWGG	
Kpn14I	CCWGG	CCWGG	
Kpn16I	CCWGG	CCWGG	
Kpn19I	CCGCGG	CCGCGG	
Kpn30I	GCGCGC	GCGCGC	
Kpn378I	CCGCGG	CCGCGG	
KpnAI	GAANNNNNNTGCC	GGCANNNNNNTTC	
M.KpnAI	GAANNNNNNTGCC	GAANNNNNNTGCC	
KpnBI	CAAANNNNNNRTCA	TGAYNNNNNNTTTG	
M.KpnBI	CAAANNNNNNRTCA	CAAANNNNNNRTCA	
KpnK14I	GGTACC	GGTACC	
Kpn2kI	CCNGG	CCNGG	
M.Kpn2kI	CCNGG	CCNGG	
Kpn49kI	GAATTC	GAATTC	
Kpn49kII	CCSGG	CCSGG	
KspI	CCGCGG	CCGCGG	MS.
Ksp22I	TGATCA	TGATCA	IV.
Ksp632I	CTCTTC	GAAGAG	M.
KspAI	GTTAAC	GTTAAC	F.
KspHK12I	CCWGG	CCWGG	
KspHK14I	CCWGG	CCWGG	
KspHK15I	YGGCCR	YGGCCR	
KteAI	CCCGGG	CCCGGG	
Kzo9I	GATC	GATC	I.
Kzo49I	GGWCC	GGWCC	
LcaI	ATCGAT	ATCGAT	
LfeI	GCAGC	GCTGC	
LguI	GCTCTTC	GAAGAGC	F.
LlaI	?	?	
I-LlaI	CACATCCATAACCATATCATTTTT	AAAAATGATATGGTTATGGATGTG	
M.LlaI	?	?	
Lla82I	?	?	
M.Lla82I	?	?	
Lla497I	CCWGG	CCWGG	
Lla1403I	?	?	
M.Lla1403I	?	?	
Lla2614I	?	?	
M.Lla2614I	?	?	
M.Lla5598I	?	?	
LlaAI	GATC	GATC	
M1.LlaAI	GATC	GATC	
M2.LlaAI	GATC	GATC	
LlaBI	CTRYAG	CTRYAG	
M.LlaBI	CTRYAG	CTRYAG	
LlaBIII	?	?	
LlaCI	AAGCTT	AAGCTT	
M.LlaCI	AAGCTT	AAGCTT	
LlaDI	AGTACT	AGTACT	
M.LlaDI	AGTACT	AGTACT	
LlaDII	GCNGC	GCNGC	
M.LlaDII	GCNGC	GCNGC	
LlaDCHI	GATC	GATC	
M1.LlaDCHI	GATC	GATC	
M2.LlaDCHI	GATC	GATC	
LlaEI	?	?	
LlaFI	?	?	

M.LlaFI	?	?	
LlaGI	?	?	
LlaG2I	GCTAGC	GCTAGC	
M1.LlaJI	GACGC	GACGC	
M2.LlaJI	GACGC	GACGC	
R1.LlaJI	?	?	
R2.LlaJI	?	?	
LlaKR2I	GATC	GATC	
M.LlaKR2I	GATC	GATC	
LlaMI	CCNGG	CCNGG	
M1.LlaMI	CCNGG	CCNGG	
M2.LlaMI	CCNGG	CCNGG	
M.LlaPI	?	?	
LldI	?	?	
M.LldI	?	?	
M.LmoA118I	?	?	
M.LmoF4565I	GATC	GATC	
Lmu60I	CCTNAGG	CCTNAGG	
LplI	ATCGAT	ATCGAT	
LpnI	RGCGCY	RGCGCY	
LpnII	?	?	
LspI	TTCGAA	TTCGAA	
Lsp1109I	GCAGC	GCTGC	
M.Lsp1109I	GCAGC	GCAGC	
Lsp1109II	GATC	GATC	
Lsp1270I	RCATGY	RCATGY	
LweI	GCATC	GATGC	F.
MabI	ACCWGGT	ACCWGGT	I.
MaeI	CTAG	CTAG	M.
MaeII	ACGT	ACGT	M.
MaeIII	GTNAC	GTNAC	M.
MaeK81I	CGTACG	CGTACG	
MaeK81II	GGNCC	GGNCC	
MalI	GATC	GATC	I.
MamI	GATNNNNATC	GATNNNNATC	M.
M.MamI	GATNNNNATC	GATNNNNATC	
MarI	AGCT	AGCT	
MauI	CTGCAG	CTGCAG	
MauAI	GCCGGC	GCCGGC	
MavI	CTCGAG	CTCGAG	
MbiI	CCGCTC	GAGCGG	F.
MboI	GATC	GATC	ABCFGKNQRUXY.
M1.MboI	GATC	GATC	
M2.MboI	GATC	GATC	
MboII	GAAGA	TCTTC	AFGIJKNOQRVX.
M1.MboII	GAAGA	GAAGA	
M2.MboII	GAAGA	GAAGA	
M.MbuI	?	?	
M.MbuII	?	?	
M.MbuIII	?	?	
M.MbuIV	?	?	
MbvI	?	?	
McaI	CTCGAG	CTCGAG	
McaAI	GGCGCC	GGCGCC	
McaBI	?	?	
McaTI	GCGCGC	GCGCGC	
M.McaTI	GCGCGC	GCGCGC	
MchI	GGCGCC	GGCGCC	
MchAI	GCGGCCGC	GCGGCCGC	
MchAII	GGCC	GGCC	
McrI	CGRYCG	CGRYCG	
MecI	CTCGAG	CTCGAG	
Mel3JI	GATC	GATC	
Mel5JI	GATC	GATC	
Mel7JI	GATC	GATC	
Mel4OI	GATC	GATC	
Mel5OI	GATC	GATC	
Mel2TI	GATC	GATC	
Mel5TI	GATC	GATC	
MeuI	GATC	GATC	
MfeI	CAATTG	CAATTG	N.
M.MfeI	CAATTG	CAATTG	
MflI	RGATCY	RGATCY	K.
MfoI	GGWCC	GGWCC	
MfoAI	GGCC	GGCC	
PI-MgaI	CGTAGCTGCCCAGTATGAGTCA	TGACTCATACTGGGCAGCTACG	
MglI	?	?	
MglII	?	?	
Mgl14481I	CCSGG	CCSGG	
MgoI	GATC	GATC	

MhaI	CTCGAG	CTCGAG	
MhaAI	CTGCAG	CTGCAG	
MhlI	GDGCHC	GDGCHC	IV.
MhoI	GGNCC	GGNCC	
Mho2111I	AGCT	AGCT	
Mho2965I	GCGC	GCGC	
MisI	GCCGGC	GCCGGC	
MizI	CTGCAG	CTGCAG	
MjaI	CTAG	CTAG	
M.MjaI	CTAG	CTAG	
MjaII	GGNCC	GGNCC	
M.MjaII	GGNCC	GGNCC	
MjaIII	GATC	GATC	
M.MjaIII	GATC	GATC	
MjaIV	GTNNAC	GTNNAC	
MjaV	GTAC	GTAC	
M.MjaV	GTAC	GTAC	
M.MjaVI	CCGG	CCGG	
MkiI	AAGCTT	AAGCTT	
MkrI	CTGCAG	CTGCAG	
MkrAI	GATC	GATC	
MlaI	TTCGAA	TTCGAA	
MlaAI	CTCGAG	CTCGAG	
MleI	GGATCC	GGATCC	
MliI	GGWCC	GGWCC	
MlsI	TGGCCA	TGGCCA	F.
MltI	AGCT	AGCT	
MluI	ACGCGT	ACGCGT	ABFGHIJKMNOQRSUVX.
M.MluI	ACGCGT	ACGCGT	
Mlu23I	GGATCC	GGATCC	
Mlu31I	TGGCCA	TGGCCA	
Mlu40I	GDGCHC	GDGCHC	
Mlu1106I	RGGWCCY	RGGWCCY	
Mlu2300I	CCWGG	CCWGG	
Mlu9273I	TCGCGA	TCGCGA	
Mlu9273II	GCCGGC	GCCGGC	
MluB2I	TCGCGA	TCGCGA	
MluCI	AATT	AATT	
MluNI	TGGCCA	TGGCCA	MS.
MlyI	GAGTC	GACTC	N.
M.MlyI	GASTC	GASTC	
Mly113I	GGCGCC	GGCGCC	I.
MmaI	CTGCAG	CTGCAG	
MmeI	TCCRAC	GTYGGA	NX.
M.MmeI	TCCRAC	TCCRAC	
MmeII	GATC	GATC	
M.MmeII	GATC	GATC	
Mmu5I	GATC	GATC	
M.Mmu5I	GATC	GATC	
M.Mmu5II	GATC	GATC	
M.MmuDnmt1	?	?	
M.MmuDnmt3A	?	?	
M.MmuDnmt3B	?	?	
MmuP2I	GATC	GATC	
MniI	GGCC	GGCC	
MniII	CCGG	CCGG	
MnlI	CCTC	GAGG	FGINQVX.
M1.MnlI	CCTC	CCTC	
M2.MnlI	CCTC	CCTC	
MnnI	GTYRAC	GTYRAC	
MnnII	GGCC	GGCC	
MnnIII	?	?	
MnnIV	GCGC	GCGC	
MnoI	CCGG	CCGG	
MnoII	?	?	
MnoIII	GATC	GATC	
MosI	GATC	GATC	
MphI	CCWGG	CCWGG	
Mph1103I	ATGCAT	ATGCAT	F.
Mph1103II	GATC	GATC	
Mpr154I	CCGCGG	CCGCGG	
MpsI	CCWGG	CCWGG	
MpuI	CTCGAG	CTCGAG	
MpuUI	?	?	
M.MpuUI	?	?	
MraI	CCGCGG	CCGCGG	
MreI	CGCCGGCG	CGCCGGCG	
MrhI	CTCGAG	CTCGAG	
MroI	TCCGGA	TCCGGA	MO.
MroNI	GCCGGC	GCCGGC	IV.

MroXI	GAANNNNTTC	GAANNNNTTC	IV.
MsaI	GGCGCC	GGCGCC	
MscI	TGGCCA	TGGCCA	BNO.
M.MscI	TGGCCA	TGGCCA	
MscAI	CTCGAG	CTCGAG	
MseI	TTAA	TTAA	BN.
M.MseI	TTAA	TTAA	
MsiI	CTCGAG	CTCGAG	
MsiII	?	?	
MslI	CAYNNNNRTG	CAYNNNNRTG	N.
M.MslI	CAYNNNNRTG	CAYNNNNRTG	
I-MsoI	CTGGGTTCAAAACGTCGTGAGACAGTTTGG	CCAAACTGTCTCACGACGTTTTGAACCCAG	
MspI	CCGG	CCGG	AFGHIJKMNOQRSUVXY.
M.MspI	CCGG	CCGG	N.
Msp11I	CTGCAG	CTGCAG	
Msp16I	TGGCCA	TGGCCA	
Msp17I	GRCGYC	GRCGYC	
Msp20I	TGGCCA	TGGCCA	IV.
Msp23I	TCTAGA	TCTAGA	
Msp23II	CTCGAG	CTCGAG	
Msp24I	GGNCC	GGNCC	
Msp67I	CCNGG	CCNGG	
Msp67II	GATC	GATC	
Msp130I	?	?	
Msp199I	CCGG	CCGG	
MspAI	GGWCC	GGWCC	
MspA1I	CMGCKG	CMGCKG	INRV.
M.MspA1I	CMGCKG	CMGCKG	
MspBI	GATC	GATC	
MspB4I	GGYRCC	GGYRCC	
MspB6I	?	?	
MspCI	CTTAAG	CTTAAG	C.
MspR9I	CCNGG	CCNGG	I.
M.MspSD10I	GACNNNGTC	GACNNNGTC	
MspSWI	ATTTAAAT	ATTTAAAT	
MspV281I	GWGCWC	GWGCWC	
MspYI	YACGTR	YACGTR	
MssI	GTTTAAAC	GTTTAAAC	F.
MstI	TGCGCA	TGCGCA	
MstII	CCTNAGG	CCTNAGG	
MthI	GATC	GATC	
Mth1047I	GATC	GATC	
MthAI	GATC	GATC	
MthBI	GGNCC	GGNCC	
MthFI	CTAG	CTAG	
M.MthFI	CTAG	CTAG	
MthTI	GGCC	GGCC	
M.MthTI	GGCC	GGCC	
MthZI	CTAG	CTAG	
M.MthZI	CTAG	CTAG	
PI-MtuI	AACGCGGTCGGCAACCGCACCCGGGTCAC	GTGACCCGGGTGCGGTTGCCGACCGCGTT	
MunI	CAATTG	CAATTG	FKM.
M.MunI	CAATTG	CAATTG	
MvaI	CCWGG	CCWGG	AFGKMOS.
M.MvaI	CCWGG	CCWGG	
Mva16I	TTCGAA	TTCGAA	
Mva1269I	GAATGC	GCATTC	F.
M.Mva1269I	GAATGC	GAATGC	
MvaAI	CGCG	CGCG	
MviI	?	?	
MviII	?	?	
Mvi80424	?	?	
MvnI	CGCG	CGCG	M.
MvrI	CGATCG	CGATCG	U.
MvsI	GGTACC	GGTACC	
MvsAI	GGTACC	GGTACC	
MvsBI	GGTACC	GGTACC	
MvsCI	GGTACC	GGTACC	
MvsDI	GGTACC	GGTACC	
MvsEI	GGTACC	GGTACC	
MwhI	GTTAAC	GTTAAC	
MwoI	GCNNNNNNNGC	GCNNNNNNNGC	N.
M.MwoI	GCNNNNNNNGC	GCNNNNNNNGC	
MxaI	GAGCTC	GAGCTC	
MziI	CAGCTG	CAGCTG	
I-NaaI	?	?	
NaeI	GCCGGC	GCCGGC	ACKMNORU.
M.NaeI	GCCGGC	GCCGGC	
NamI	GGCGCC	GGCGCC	
NanI	GATATC	GATATC	

I-NanI	AAGTCTGGTGCCAGCACCCGC	GCGGGTGCTGGCACCAGACTT	
NanII	GATC	GATC	
NarI	GGCGCC	GGCGCC	GJMNOQRUX.
NasI	CTGCAG	CTGCAG	
NasBI	GGATCC	GGATCC	
NasSI	GAGCTC	GAGCTC	
NasWI	GCCGGC	GCCGGC	
NbaI	GCCGGC	GCCGGC	
NblI	CGATCG	CGATCG	
NbrI	GCCGGC	GCCGGC	
NcaI	GANTC	GANTC	
NciI	CCSGG	CCSGG	GJNORS.
NciAI	GATC	GATC	
NcoI	CCATGG	CCATGG	ABCFGHJKMNOQRSUXY.
M.NcoI	CCATGG	CCATGG	
NcrI	AGATCT	AGATCT	
M.NcrNI	?	?	
M.NcrNII	?	?	
NcuI	GAAGA	TCTTC	
M1.NcuI	GAAGA	GAAGA	
NcuII	CCCG	CGGG	
NdaI	GGCGCC	GGCGCC	
NdeI	CATATG	CATATG	ABFGJKMNRSXY.
M.NdeI	CATATG	CATATG	
NdeII	GATC	GATC	GJMRS.
M.NdeII	GATC	GATC	
NflI	GATC	GATC	
NflII	?	?	
NflIII	?	?	
NflAI	GATATC	GATATC	
NflAII	GATC	GATC	
NflBI	GATC	GATC	
NgbI	CTGCAG	CTGCAG	
NgoAI	RGCGCY	RGCGCY	
M.NgoAI	RGCGCY	RGCGCY	
NgoAII	GGCC	GGCC	
M.NgoAII	GGCC	GGCC	
NgoAIII	CCGCGG	CCGCGG	
M.NgoAIII	CCGCGG	CCGCGG	
NgoAIV	GCCGGC	GCCGGC	
M.NgoAIV	GCCGGC	GCCGGC	
NgoAV	GCANNNNNNNNTGC	GCANNNNNNNNTGC	
NgoAV-1	?	?	
M.NgoAV	GCANNNNNNNNTGC	GCANNNNNNNNTGC	
M.NgoAV-1	?	?	
NgoBI	RGCGCY	RGCGCY	
M.NgoBI	RGCGCY	RGCGCY	
M.NgoBII	GGCC	GGCC	
NgoBV	GGNNCC	GGNNCC	
M.NgoBV	GGNNCC	GGNNCC	
NgoBVIII	GGTGA	TCACC	
M1.NgoBVIII	GGTGA	GGTGA	
M2.NgoBVIII	GGTGA	GGTGA	
M.NgoBIX	GTANNNNNCTC	GTANNNNNCTC	
M.NgoBXII	GCNGC	GCNGC	
NgoCI	RGCGCY	RGCGCY	
NgoCII	GGCC	GGCC	
NgoDI	?	?	
M.NgoDI	?	?	
NgoDIII	CCGCGG	CCGCGG	
M.NgoDIII	CCGCGG	CCGCGG	
NgoDVIII	GGTGA	TCACC	
NgoDXIV	GATC	GATC	
M.NgoEI	RGCGCY	RGCGCY	
NgoEII	GCGC	GCGC	
NgoFVII	GCSGC	GCSGC	
M.NgoFVII	GCSGC	GCSGC	
NgoGI	RGCGCY	RGCGCY	
M.NgoGI	RGCGCY	RGCGCY	
M.NgoGII	GGCC	GGCC	
NgoGIII	CCGCGG	CCGCGG	
M.NgoGIII	CCGCGG	CCGCGG	
NgoGV	GGNNCC	GGNNCC	
M.NgoGV	GGNNCC	GGNNCC	
M.NgoHVIII	GGTGA	GGTGA	
NgoJI	RGCGCY	RGCGCY	
NgoJIII	CCGCGG	CCGCGG	
NgoJVIII	GGTGA	TCACC	
NgoKIII	CCGCGG	CCGCGG	
M.NgoLII	GGCC	GGCC	

NgoMI	RGCGCY	RGCGCY	
M.NgoMI	RGCGCY	RGCGCY	
M.NgoMII	GGCC	GGCC	
NgoMIII	CCGCGG	CCGCGG	
M.NgoMIII	CCGCGG	CCGCGG	
NgoMIV	GCCGGC	GCCGGC	NR.
M.NgoMIV	GCCGGC	GCCGGC	
M.NgoMV	GGNNCC	GGNNCC	
NgoMVIII	GGTGA	TCACC	
M.NgoMVIII	GGTGA	GGTGA	
NgoMX	?	?	
M.NgoMX	?	?	
M.NgoMXV	GCCHR	GCCHR	
NgoNII	GGCC	GGCC	
M.NgoNII	GGCC	GGCC	
NgoPII	GGCC	GGCC	
M.NgoPII	GGCC	GGCC	
NgoPIII	CCGCGG	CCGCGG	
M.NgoPIII	CCGCGG	CCGCGG	
NgoSII	GGCC	GGCC	
M.NgoSII	GGCC	GGCC	
NgoTII	GGCC	GGCC	
M.NgoTII	GGCC	GGCC	
NgoWI	RGCGCY	RGCGCY	
NheI	GCTAGC	GCTAGC	ABFGJKMNORSU.
M.NheI	GCTAGC	GCTAGC	
I-NitI	AAGTCTGGTGCCAGCACCCGC	GCGGGTGCTGGCACCAGACTT	
I-NjaI	AAGTCTGGTGCCAGCACCCGC	GCGGGTGCTGGCACCAGACTT	
NlaI	GGCC	GGCC	
M.NlaI	GGCC	GGCC	
NlaII	GATC	GATC	
NlaIII	CATG	CATG	GN.
M.NlaIII	CATG	CATG	
NlaIV	GGNNCC	GGNNCC	GN.
M.NlaIV	GGNNCC	GGNNCC	
NlaX	CCNGG	CCNGG	
M.NlaX	CCNGG	CCNGG	
NlaDI	GATC	GATC	
NlaDII	GGNCC	GGNCC	
NlaDIII	CCGCGG	CCGCGG	
NlaSI	CCGCGG	CCGCGG	
NlaSII	GRCGYC	GRCGYC	
NliI	CYCGRG	CYCGRG	
NliII	GGWCC	GGWCC	
Nli3877I	CYCGRG	CYCGRG	
Nli3877II	GGWCC	GGWCC	
M.NmaPhiCh1I	GATC	GATC	
NmeI	?	?	
NmeII	?	?	
NmeIII	?	?	
NmeIV	?	?	
M.NmeAI	CCGG	CCGG	
NmeAII	GATC	GATC	
NmeBI	GACGC	GCGTC	
M1.NmeBI	GACGC	GACGC	
M2.NmeBI	GACGC	GACGC	
NmeBL859I	GATC	GATC	
NmeCI	GATC	GATC	
M.NmeDI	RCCGGB	RCCGGB	
NmeRI	CAGCTG	CAGCTG	
NmeSI	AGTACT	AGTACT	
M.NmeSI	AGTACT	AGTACT	
NmiI	GGTACC	GGTACC	
NmuI	GCCGGC	GCCGGC	
NmuAI	CYCGRG	CYCGRG	
NmuAII	GGWCC	GGWCC	
NmuCI	GTSAC	GTSAC	F.
NmuDI	GATC	GATC	
NmuEI	GATC	GATC	
NmuEII	GGNCC	GGNCC	
NmuFI	GCCGGC	GCCGGC	
NmuSI	GGNCC	GGNCC	
NocI	CTGCAG	CTGCAG	
NopI	GTCGAC	GTCGAC	
NopII	?	?	
NotI	GCGGCCGC	GCGGCCGC	ABCFGHJKMNOQRSUXY.
M.NotI	GCGGCCGC	GCGGCCGC	
NovI	?	?	
NovII	GANTC	GANTC	
NpeBY1I	?	?	

NpeHEMI	?	?	
NpeHKVVI	?	?	
NphI	GATC	GATC	
NruI	TCGCGA	TCGCGA	ABCGIJKMNOQRSUX.
M.NruI	TCGCGA	TCGCGA	
NruGI	GACNNNNNGTC	GACNNNNNGTC	
NsbI	TGCGCA	TGCGCA	FK.
NsiI	ATGCAT	ATGCAT	BGHJMNRSU.
M.NsiI	ATGCAT	ATGCAT	
NsiAI	GATC	GATC	
NsiCI	GATATC	GATATC	
NsiHI	GANTC	GANTC	
NspI	RCATGY	RCATGY	MN.
M.NspI	RCATGY	RCATGY	
NspII	GDGCHC	GDGCHC	
NspIII	CYCGRG	CYCGRG	
M.NspIII	CYCGRG	CYCGRG	
NspIV	GGNCC	GGNCC	
NspV	TTCGAA	TTCGAA	JO.
M.NspV	TTCGAA	TTCGAA	
Nsp152I	?	?	
Nsp7121I	GGNCC	GGNCC	
Nsp29132I	TTCGAA	TTCGAA	
Nsp29132II	GGATCC	GGATCC	
NspAI	GATC	GATC	
NspBI	TTCGAA	TTCGAA	
NspBII	CMGCKG	CMGCKG	
NspDI	CYCGRG	CYCGRG	
NspDII	GGWCC	GGWCC	
NspEI	CYCGRG	CYCGRG	
NspEII	?	?	
NspFI	TTCGAA	TTCGAA	
NspGI	GGWCC	GGWCC	
NspHI	RCATGY	RCATGY	
M.NspHI	RCATGY	RCATGY	
NspHII	GGWCC	GGWCC	
NspHIII	TGCGCA	TGCGCA	
NspJI	TTCGAA	TTCGAA	
NspKI	GGWCC	GGWCC	
NspLI	TGCGCA	TGCGCA	
NspLII	GGNCC	GGNCC	
NspLIII	?	?	
NspLIV	?	?	
NspLKI	GGCC	GGCC	
NspMI	TGCGCA	TGCGCA	
NspMACI	AGATCT	AGATCT	
NspSAI	CYCGRG	CYCGRG	
NspSAII	GGTNACC	GGTNACC	
NspSAIII	CCATGG	CCATGG	
NspSAIV	GGATCC	GGATCC	
NspWI	GCCGGC	GCCGGC	
NsuI	GATC	GATC	
NsuDI	GATC	GATC	
NtaI	GACNNNGTC	GACNNNGTC	
NtaSI	AGGCCT	AGGCCT	
NtaSII	GCCGGC	GCCGGC	
M.NtbDRM1	?	?	
NunI	?	?	
NunII	GGCGCC	GGCGCC	
OchI	GGCC	GGCC	
OcoI	CTCGAG	CTCGAG	
OfoI	CYCGRG	CYCGRG	
OkrAI	GGATCC	GGATCC	
M.OkrAI	GGATCC	GGATCC	
OliI	CACNNNNGTG	CACNNNNGTG	F.
OmiAI	GRGCYC	GRGCYC	
OmiBI	GRGCYC	GRGCYC	
OmiBII	GTMKAC	GTMKAC	
M.OsaDnmt1-1	?	?	
M.OsaDnmt1-2	?	?	
OspI	TTCGAA	TTCGAA	
OtuI	AGCT	AGCT	
OtuNI	AGCT	AGCT	
OxaI	AGCT	AGCT	
OxaII	?	?	
OxaNI	CCTNAGG	CCTNAGG	
PabI	GTAC	GTAC	
M.PabI	GTAC	GTAC	
PI-PabI	GGGGGCAGCCAGTGGTCCCGTT	AACGGGACCACTGGCTGCCCCC	
PI-PabII	ACCCCTGTGGAGAGGAGCCCCTC	GAGGGGCTCCTCTCCACAGGGGT	

PacI	TTAATTAA	TTAATTAA	GNO.
Pac25I	CCCGGG	CCCGGG	
M.Pac25I	CCCGGG	CCCGGG	
Pac1110I	GGATCC	GGATCC	
Pac1110II	GATATC	GATATC	
PaeI	GCATGC	GCATGC	F.
M.PaeI	GCATGC	GCATGC	
Pae7I	CCGCGG	CCGCGG	
Pae8I	CTGCAG	CTGCAG	
Pae9I	CTGCAG	CTGCAG	
Pae14I	CTGCAG	CTGCAG	
Pae15I	CTGCAG	CTGCAG	
Pae17I	CCGCGG	CCGCGG	
Pae22I	CTGCAG	CTGCAG	
Pae24I	CTGCAG	CTGCAG	
Pae25I	CTGCAG	CTGCAG	
Pae26I	CTGCAG	CTGCAG	
Pae36I	CCGCGG	CCGCGG	
Pae39I	CTGCAG	CTGCAG	
Pae40I	CTGCAG	CTGCAG	
Pae41I	CTGCAG	CTGCAG	
Pae42I	CCGCGG	CCGCGG	
Pae43I	CCGCGG	CCGCGG	
Pae44I	CCGCGG	CCGCGG	
Pae177I	GGATCC	GGATCC	
Pae181I	CCSGG	CCSGG	
PaeAI	CCGCGG	CCGCGG	
PaeBI	CCCGGG	CCCGGG	
PaeCI	GCATGC	GCATGC	
PaeHI	GRGCYC	GRGCYC	
PaePI	CTGCAG	CTGCAG	
PaeQI	CCGCGG	CCGCGG	
PaeR7I	CTCGAG	CTCGAG	N.
M.PaeR7I	CTCGAG	CTCGAG	
Pae2kI	AGATCT	AGATCT	
Pae5kI	CCGCGG	CCGCGG	
Pae14kI	CCGCGG	CCGCGG	
Pae17kI	CAGCTG	CAGCTG	
Pae18kI	AGATCT	AGATCT	
PagI	TCATGA	TCATGA	F.
PaiI	GGCC	GGCC	
I-PakI	CTGGGTTCAAAACGTCGTGAGACAGTTTGG	CCAAACTGTCTCACGACGTTTTGAACCCAG	
PalI	GGCC	GGCC	
PalAI	GGCGCGCC	GGCGCGCC	I.
PamI	TGCGCA	TGCGCA	
PamII	GRCGYC	GRCGYC	
PanI	CTCGAG	CTCGAG	
ParI	TGATCA	TGATCA	
PasI	CCCWGGG	CCCWGGG	F.
PatAI	GGCGCC	GGCGCC	
PauI	GCGCGC	GCGCGC	F.
PauAI	RCATGY	RCATGY	
PauAII	TTTAAA	TTTAAA	
PbrTI	GATC	GATC	
PbuJKI	GGATG	CATCC	
PbuMZI	ATTAAT	ATTAAT	
Pca17AI	CCWGG	CCWGG	
PceI	AGGCCT	AGGCCT	IV.
PciI	ACATGT	ACATGT	IN.
PciSI	GCTCTTC	GAAGAGC	I.
PctI	GAATGC	GCATTC	IV.
I-PcuAI	?	?	
I-PcuVI	?	?	
Pde12I	GGNCC	GGNCC	
Pde133I	GGCC	GGCC	
Pde137I	CCGG	CCGG	
PdiI	GCCGGC	GCCGGC	F.
PdmI	GAANNNNTTC	GAANNNNTTC	F.
Pei9403I	GATC	GATC	
PfaI	GATC	GATC	
PfaAI	GGYRCC	GGYRCC	
PfaAII	CATATG	CATATG	
PfaAIII	GCATGC	GCATGC	
PfeI	GAWTC	GAWTC	F.
PflI	?	?	
Pfl8I	GGATCC	GGATCC	
Pfl16I	GATATC	GATATC	
Pfl18I	GAGCTC	GAGCTC	
Pfl19I	GGWCC	GGWCC	
Pfl21I	CTGCAG	CTGCAG	

Pfl23I	GTGCAC	GTGCAC	
Pfl23II	CGTACG	CGTACG	F.
Pfl27I	RGGWCCY	RGGWCCY	
Pfl37I	CTGCAG	CTGCAG	
Pfl67I	CTCGAG	CTCGAG	
Pfl1108I	TCGTAG	CTACGA	
Pfl1108II	CCGCGG	CCGCGG	
PflAI	CGCG	CGCG	
PflBI	CCANNNNNTGG	CCANNNNNTGG	
PflFI	GACNNNGTC	GACNNNGTC	N.
PflKI	GGCC	GGCC	
PflMI	CCANNNNNTGG	CCANNNNNTGG	N.
M.PflMI	CCANNNNNTGG	CCANNNNNTGG	
PflNI	CTCGAG	CTCGAG	
PflWI	CTCGAG	CTCGAG	
PfoI	TCCNGGA	TCCNGGA	F.
Pfr12I	GTGCAC	GTGCAC	
PI-PfuI	GAAGATGGGAGGAGGGACCGGACTCAACTT	AAGTTGAGTCCGGTCCCTCCTCCCATCTTC	
PI-PfuII	ACGAATCCATGTGGAGAAGAGCCTCTATA	TATAGAGGCTCTTCTCCACATGGATTCGT	
PfuNI	CGTACG	CGTACG	
PgaI	ATCGAT	ATCGAT	
M.PgiI	GATC	GATC	
PglI	GCCGGC	GCCGGC	
PglII	?	?	
Pgl34I	CACGTG	CACGTG	
PhaI	GCATC	GATGC	
M.PhaI	GCATC	GCATC	
PhaAI	?	?	
M.PhaAI	?	?	
PhaBI	?	?	
M.PhaBI	?	?	
M.PhaTDam	GATC	GATC	
M.PhiBssHII	ACGCGT	ACGCGT	
M.PhiBssHII	CCGCGG	CCGCGG	
M.PhiBssHII	RGCGCY	RGCGCY	
M.PhiBssHII	RCCGGY	RCCGGY	
M.PhiBssHII	GCGCGC	GCGCGC	
M.PhiHII	?	?	
M.PhiMx8I	CTSSAG	CTSSAG	
M.Phi3TI	GGCC	GGCC	
M.Phi3TI	GCNGC	GCNGC	
M.Phi3TII	TCGA	TCGA	
F-PhiU5I	AATAACCTGAAGTATCAATC	GATTGATACTTCAGGTTATT	
PhoI	GGCC	GGCC	N.
M.PhoI	GGCC	GGCC	
M.PhoII	GATC	GATC	
PinI	AGTACT	AGTACT	
PinAI	ACCGGT	ACCGGT	BM.
PinBI	ATGCAT	ATGCAT	
PinBII	TCCGGA	TCCGGA	
PI-PkoI	GATTTTAGATCCCTGTACC	GGTACAGGGATCTAAAATC	
PI-PkoII	CAGTACTACGGTTAC	GTAACCGTAGTACTG	
PlaI	GGCC	GGCC	
PlaII	TTCGAA	TTCGAA	
PlaAI	CYCGRG	CYCGRG	
PlaAII	GTAC	GTAC	
PleI	GAGTC	GACTC	N.
M.PleI	GAGTC	GAGTC	
Ple19I	CGATCG	CGATCG	I.
Ple214I	GGCC	GGCC	
PliI	GTGCAC	GTGCAC	
M.PliMCDnmt1	?	?	
PluI	AGGCCT	AGGCCT	
PmaI	CTGCAG	CTGCAG	
Pma44I	CTGCAG	CTGCAG	
PmaCI	CACGTG	CACGTG	AK.
PmeI	GTTTAAAC	GTTTAAAC	GN.
Pme35I	CCGG	CCGG	
Pme55I	AGGCCT	AGGCCT	
PmiI	?	?	
PmlI	CACGTG	CACGTG	N.
PmnI	GGCGCC	GGCGCC	
M.PmuADam	GATC	GATC	
M.PmuDam	GATC	GATC	
PmyI	CTGCAG	CTGCAG	
PntI	CGATCG	CGATCG	
I-PogI	CTTCAGTATGCCCCGAAAC	GTTTCGGGGCATACTGAAG	
PolI	GGWCC	GGWCC	
I-PorI	GCGAGCCCGTAAGGGTGTGTACGGG	CCCGTACACACCCTTACGGGCTCGC	
PovI	TGATCA	TGATCA	

PpaI	GGTCTC	GAGACC	
PpaAI	TTCGAA	TTCGAA	
PpaAII	TCGA	TCGA	
PpeI	GGGCCC	GGGCCC	
Pph14I	GGYRCC	GGYRCC	
Pph288I	GATC	GATC	
Pph1579I	GGNCC	GGNCC	
Pph1591I	?	?	
Pph1773I	GGNCC	GGNCC	
Pph2059I	CTGCAG	CTGCAG	
Pph2066I	CTGCAG	CTGCAG	
Pph3215I	GWGCWC	GWGCWC	
PpiI	GAACNNNNNCTC	GAGNNNNNGTTC	F.
PpiI	GAGNNNNNGTTC	GAACNNNNNCTC	F.
I-PpoI	TAACTATGACTCTCTTAAGGTAGCCAAAT	ATTTGGCTACCTTAAGAGAGTCATAGTTA	R.
PpsI	GAGTC	GACTC	I.
PpuI	GGCC	GGCC	
Ppu6I	YACGTR	YACGTR	
Ppu10I	ATGCAT	ATGCAT	
Ppu11I	YACGTR	YACGTR	
Ppu13I	AGGCCT	AGGCCT	
Ppu20I	GRGCYC	GRGCYC	
Ppu21I	YACGTR	YACGTR	F.
M.Ppu21I	YACGTR	YACGTR	
Ppu111I	GAATTC	GAATTC	
M.Ppu111I	GAATTC	GAATTC	
Ppu1253I	GACGTC	GACGTC	
M.Ppu1253I	GACGTC	GACGTC	
PpuAI	CGTACG	CGTACG	
PpuMI	RGGWCCY	RGGWCCY	NO.
M.PpuMI	RGGWCCY	RGGWCCY	
PpuXI	RGGWCCY	RGGWCCY	
Pru2I	GGCC	GGCC	
M.PsaDnmt1	?	?	
Psb9879I	GGCC	GGCC	
PscI	ACATGT	ACATGT	F.
Psc2I	GAANNNNTTC	GAANNNNTTC	
Psc2II	?	?	
Psc18I	?	?	
Psc27I	TTCGAA	TTCGAA	
Psc28I	TTCGAA	TTCGAA	
Psc45I	?	?	
Psc49I	?	?	
Psc97I	?	?	
Psc126I	?	?	
Psc128I	?	?	
Psc193I	?	?	
PseI	GGNCC	GGNCC	
PshAI	GACNNNNGTC	GACNNNNGTC	AKN.
M.PshAI	GACNNNNGTC	GACNNNNGTC	
PshBI	ATTAAT	ATTAAT	K.
PshCI	CACGTG	CACGTG	
PshDI	CACGTG	CACGTG	
PshEI	CTGCAG	CTGCAG	
PsiI	TTATAA	TTATAA	IN.
PspI	GGNCC	GGNCC	
PI-PspI	TGGCAAACAGCTATTATGGGTATTATGGGT	ACCCATAATACCCATAATAGCTGTTTGCCA	N.
Psp03I	GGWCC	GGWCC	
Psp3I	CAGCTG	CAGCTG	
Psp4I	CTCGAG	CTCGAG	
Psp5I	CAGCTG	CAGCTG	
Psp5II	RGGWCCY	RGGWCCY	F.
Psp6I	CCWGG	CCWGG	I.
Psp23I	CTGCAG	CTGCAG	
Psp28I	CTGCAG	CTGCAG	
Psp29I	GGCC	GGCC	
Psp30I	GGGCCC	GGGCCC	
Psp31I	GRGCYC	GRGCYC	
Psp32I	GTCGAC	GTCGAC	
Psp33I	GTCGAC	GTCGAC	
Psp38I	CACGTG	CACGTG	
Psp39I	CCWGG	CCWGG	
Psp46I	CTGCAG	CTGCAG	
Psp56I	GGATCC	GGATCC	
Psp61I	GCCGGC	GCCGGC	
Psp89I	GTCGAC	GTCGAC	
Psp1406I	AACGTT	AACGTT	FKM.
PspAI	CCCGGG	CCCGGG	
PspALI	CCCGGG	CCCGGG	
PspBI	CACGTG	CACGTG	

Psp124BI	GAGCTC	GAGCTC	IV.
PspCI	CACGTG	CACGTG	IV.
PspDI	TCGCGA	TCGCGA	
PspEI	GGTNACC	GGTNACC	IV.
PspGI	CCWGG	CCWGG	N.
M.PspGI	CCWGG	CCWGG	
PspLI	CGTACG	CGTACG	I.
PspNI	CTCGAG	CTCGAG	
PspN4I	GGNNCC	GGNNCC	I.
PspOMI	GGGCCC	GGGCCC	INV.
PspPI	GGNCC	GGNCC	
M.PspPI	GGNCC	GGNCC	
PspPPI	RGGWCCY	RGGWCCY	I.
PspSI	CTGCAG	CTGCAG	
PspXI	VCTCGAGB	VCTCGAGB	IN.
PsrI	GAACNNNNNNTAC	GTANNNNNNGTTC	I.
PsrI	GTANNNNNNGTTC	GAACNNNNNNTAC	I.
PssI	RGGNCCY	RGGNCCY	
PssII	?	?	
PstI	CTGCAG	CTGCAG	ABCFGHIJKMNOQRSUVXY.
M.PstI	CTGCAG	CTGCAG	
PstII	CTGATG	CATCAG	
M.PstII	CTGATG	CTGATG	
PstNHI	GCTAGC	GCTAGC	
PsuI	RGATCY	RGATCY	F.
Psu161I	CGATCG	CGATCG	
PsuAI	YACGTR	YACGTR	
PsuNI	CRCCGGYG	CRCCGGYG	
M.PsuNI	?	?	
PsyI	GACNNNGTC	GACNNNGTC	F.
PtaI	TCCGGA	TCCGGA	
Pun14627I	TGCGCA	TGCGCA	
Pun14627II	CAGCTG	CAGCTG	
PunAI	CYCGRG	CYCGRG	
PunAII	RCATGY	RCATGY	
PvuI	CGATCG	CGATCG	ABFGKMNOQRSUXY.
M.PvuI	CGATCG	CGATCG	
PvuII	CAGCTG	CAGCTG	ABCFGHIJKMNOQRSUVXY.
M.PvuII	CAGCTG	CAGCTG	
Pvu84I	CGATCG	CGATCG	
Pvu84II	CAGCTG	CAGCTG	
PvuHKUI	CAGCTG	CAGCTG	
PxyARI	GATATC	GATATC	
PxyJKI	ATGCAT	ATGCAT	
PxyMZI	CCTNAGG	CCTNAGG	
Ral8I	GGATC	GATCC	
RalF40I	GATC	GATC	
RcaI	TCATGA	TCATGA	M.
RflFI	GTCGAC	GTCGAC	
M.RflFI	?	?	
RflFII	AGTACT	AGTACT	
RgaI	GCGATCGC	GCGATCGC	I.
RhcI	TCATGA	TCATGA	
RheI	GTCGAC	GTCGAC	
M.Rho11sI	GGCC	GGCC	
M.Rho11sI	GCNGC	GCNGC	
M.Rho11sII	TCGA	TCGA	
RhpI	GTCGAC	GTCGAC	
RhpII	?	?	
RhsI	GGATCC	GGATCC	
M.RhvI	?	?	
RleI	?	?	
Rle69I	GGTCTC	GAGACC	
RleAI	CCCACA	TGTGGG	
M.Rle39BI	CTGCAG	CTGCAG	
RluI	GCCGGC	GCCGGC	
Rlu1I	GATC	GATC	
Rlu3I	GGNNCC	GGNNCC	
Rlu4I	GGATCC	GGATCC	
RmaI	CTAG	CTAG	
Rma376I	TTCGAA	TTCGAA	
Rma485I	CTAG	CTAG	
Rma486I	CTAG	CTAG	
Rma490I	CTAG	CTAG	
Rma495I	CTAG	CTAG	
Rma495II	GATATC	GATATC	
Rma496I	CTAG	CTAG	
Rma496II	GATATC	GATATC	
Rma497I	CTAG	CTAG	
Rma497II	GATATC	GATATC	

Rma500I	CTAG	CTAG	
Rma501I	CTAG	CTAG	
Rma503I	CTAG	CTAG	
Rma506I	CTAG	CTAG	
Rma509I	CTAG	CTAG	
Rma510I	CTAG	CTAG	
Rma515I	CTAG	CTAG	
Rma516I	CTAG	CTAG	
Rma517I	CTAG	CTAG	
Rma518I	CTAG	CTAG	
Rma519I	CTAG	CTAG	
Rma522I	CTAG	CTAG	
Rma523I	TTCGAA	TTCGAA	
RmeI	?	?	
Rme21I	ATCGAT	ATCGAT	
M.RmeADam	GATC	GATC	
M.RnoDnmt1	?	?	
M.RraDnmtI	?	?	
RrbI	?	?	
RrhI	GTCGAC	GTCGAC	
RrhII	?	?	
Rrh4273I	GTCGAC	GTCGAC	
M.Rrh4273I	GTCGAC	GTCGAC	
RroI	GTCGAC	GTCGAC	
RruAI	?	?	
RsaI	GTAC	GTAC	BCFGHIJMNOQRSVXY.
M.RsaI	GTAC	GTAC	
RshI	CGATCG	CGATCG	
M.RshI	CGATCG	CGATCG	
RshII	CCSGG	CCSGG	
M.RshIII	GANTC	GANTC	
RspI	CGATCG	CGATCG	
RspLKI	GCATGC	GCATGC	
RspLKII	GGATCC	GGATCC	
RspXI	TCATGA	TCATGA	
RsrI	GAATTC	GAATTC	
M.RsrI	GAATTC	GAATTC	
RsrII	CGGWCCG	CGGWCCG	MNQX.
M.RsrII	CGGWCCG	CGGWCCG	
Rsr2I	CGGWCCG	CGGWCCG	I.
RtrI	GTCGAC	GTCGAC	
Rtr20I	GAAGAC	GTCTTC	
Rtr63I	GTCGAC	GTCGAC	
M.SPBetaI	GGCC	GGCC	
M.SPBetaI	GCNGC	GCNGC	
M.SPRI	GGCC	GGCC	
M.SPRI	CCGG	CCGG	
M.SPRI	CCWGG	CCWGG	
SaaI	CCGCGG	CCGCGG	
SabI	CCGCGG	CCGCGG	
SacI	GAGCTC	GAGCTC	AFGHJKMNOQRSUX.
M.SacI	GAGCTC	GAGCTC	
SacII	CCGCGG	CCGCGG	AGHJKNOQRX.
M.SacII	CCGCGG	CCGCGG	
SacIII	?	?	
SacAI	GCCGGC	GCCGGC	
SacNI	GRGCYC	GRGCYC	
SagI	GGCC	GGCC	
Sag16I	CTGCAG	CTGCAG	
M.Sag16I	CTGCAG	CTGCAG	
Sag23I	CTGCAG	CTGCAG	
M.Sag23I	CTGCAG	CTGCAG	
SaiI	GGGTC	GACCC	
SakI	CCGCGG	CCGCGG	
SalI	GTCGAC	GTCGAC	ABCFGHIJKMNOQRSUVXY.
M.SalI	GTCGAC	GTCGAC	
SalII	?	?	
Sal13I	CTGCAG	CTGCAG	
Sal1974I	CTCGAG	CTCGAG	
SalAI	GATC	GATC	
SalCI	GCCGGC	GCCGGC	
SalDI	TCGCGA	TCGCGA	
SalHI	GATC	GATC	
SalPI	CTGCAG	CTGCAG	
SanI	?	?	
SanDI	GGGWCCC	GGGWCCC	E.
SaoI	GCCGGC	GCCGGC	
SapI	GCTCTTC	GAAGAGC	N.
M1.SapI	GCTCTTC	GCTCTTC	
M2.SapI	GCTCTTC	GCTCTTC	

SarI	AGGCCT	AGGCCT	
SatI	GCNGC	GCNGC	F.
SauI	CCTNAGG	CCTNAGG	
Sau2I	GGNCC	GGNCC	
Sau5I	GGNCC	GGNCC	
Sau10I	GGTACC	GGTACC	
Sau12I	GGTCTC	GAGACC	
Sau13I	GGNCC	GGNCC	
Sau14I	GGNCC	GGNCC	
Sau15I	GATC	GATC	
Sau16I	CCWGG	CCWGG	
Sau17I	GGNCC	GGNCC	
Sau32I	GGNCC	GGNCC	
M.Sau32I	GGNCC	GGNCC	
Sau33I	GGNCC	GGNCC	
M.Sau33I	GGNCC	GGNCC	
Sau42I	?	?	
Sau90I	CTYRAG	CTYRAG	
M.Sau90I	CTYRAG	CTYRAG	
Sau93I	CTYRAG	CTYRAG	
M.Sau93I	CTYRAG	CTYRAG	
Sau96I	GGNCC	GGNCC	GJMNOU.
M.Sau96I	GGNCC	GGNCC	
Sau98I	CTYRAG	CTYRAG	
M.Sau98I	CTYRAG	CTYRAG	
Sau557I	GGNCC	GGNCC	
Sau3239I	CTCGAG	CTCGAG	
M.Sau3239I	CTCGAG	CTCGAG	
Sau6782I	GATC	GATC	
M.Sau6782I	GATC	GATC	
Sau22201I	?	?	
SauAI	GCCGGC	GCCGGC	
Sau3AI	GATC	GATC	AGHJKMNOQRSUX.
M.Sau3AI	GATC	GATC	
SauBI	GGNCC	GGNCC	
SauBMKI	GCCGGC	GCCGGC	
SauCI	GATC	GATC	
SauDI	GATC	GATC	
SauEI	GATC	GATC	
SauFI	GATC	GATC	
SauGI	GATC	GATC	
SauHI	CCTNAGG	CCTNAGG	
SauHPI	GCCGGC	GCCGGC	
SauLPI	GCCGGC	GCCGGC	
M.SauLPI	GCCGGC	GCCGGC	
SauLPII	CTCGAG	CTCGAG	
SauMI	GATC	GATC	
SauNI	GCCGGC	GCCGGC	
SauSI	GCCGGC	GCCGGC	
SauS2I	?	?	
Sau96mI	CTYRAG	CTYRAG	
M.Sau96mI	CTYRAG	CTYRAG	
SbaI	CAGCTG	CAGCTG	
M.SbaI	CAGCTG	CAGCTG	
SbfI	CCTGCAGG	CCTGCAGG	INV.
M.SbfI	CCTGCAGG	CCTGCAGG	
Sbi68I	CTCGAG	CTCGAG	
SblAI	CCWWGG	CCWWGG	
SblBI	CCWWGG	CCWWGG	
SblCI	CCWWGG	CCWWGG	
SboI	CCGCGG	CCGCGG	
Sbo13I	TCGCGA	TCGCGA	
M.Sbo13I	TCGCGA	TCGCGA	
SbrI	?	?	
SbvI	GGCC	GGCC	
ScaI	AGTACT	AGTACT	ABCFGJKMNOQRSX.
I-ScaI	TGTCACATTGAGGTGCACTAGTTATTAC	GTAATAACTAGTGCACCTCAATGTGACA	
M.ScaI	AGTACT	AGTACT	
PI-ScaI	TAAGTCGGGTGCGGAGAAAGAGGAAAAGAG	CTCTTTTCCTCTTTCTCCGCACCCGACTTA	
Sca1827I	CTCGAG	CTCGAG	
F-SceI	GATGCTGTAGGCATAGGCTTGGTT	AACCAAGCCTATGCCTACAGCATC	
I-SceI	AGTTACGCTAGGGATAACAGGGTAATATAG	CTATATTACCCTGTTATCCCTAGCGTAACT	FN.
PI-SceI	ATCTATGTCGGGTGCGGAGAAAGAGGTAAT	ATTACCTCTTTCTCCGCACCCGACATAGAT	N.
F-SceII	CTTTCCGCAACAGTAAAATT	AATTTTACTGTTGCGGAAAG	
I-SceII	TTTTGATTCTTTGGTCACCCTGAAGTATA	TATACTTCAGGGTGACCAAAGAATCAAAA	
SceIII	GCCGGC	GCCGGC	
I-SceIII	ATTGGAGGTTTTGGTAACTATTTATTACC	GGTAATAAATAGTTACCAAAACCTCCAAT	
I-SceIV	TCTTTTCTCTTGATTAGCCCTAATCTACG	CGTAGATTAGGGCTAATCAAGAGAAAAGA	
I-SceV	AATAATTTTCTTCTTAGTAATGCC	GGCATTACTAAGAAGAAAATTATT	
I-SceVI	GTTATTTAATGTTTTAGTAGTTGG	CCAACTACTAAAACATTAAATAAC	

I-SceVII	TGTCACATTGAGGTGCACTAGTTATTAC	GTAATAACTAGTGCACCTCAATGTGACA	
SceAI	CGCG	CGCG	
Scg2I	CCWGG	CCWGG	
SchI	GAGTC	GACTC	F.
SchZI	CCGCGG	CCGCGG	
SciI	CTCGAG	CTCGAG	
Sci1831I	CTCGAG	CTCGAG	
SciAI	GGTNACC	GGTNACC	
SciAII	CAGCTG	CAGCTG	
SciBI	CTCGAG	CTCGAG	
SciNI	GCGC	GCGC	
SciRI	?	?	
ScoI	GAGCTC	GAGCTC	
ScoAI	CTGCAG	CTGCAG	
ScoNI	GTGCAC	GTGCAC	
ScrFI	CCNGG	CCNGG	JMNOS.
M1.ScrFI	CCNGG	CCNGG	
M2.ScrFI	CCNGG	CCNGG	
ScuI	CTCGAG	CTCGAG	
SdaI	CCTGCAGG	CCTGCAGG	F.
SdiI	GGCCNNNNNGGCC	GGCCNNNNNGGCC	
SdiAI	CTCGAG	CTCGAG	
SduI	GDGCHC	GDGCHC	F.
M.SduI	GDGCHC	GDGCHC	
SdyI	GGNCC	GGNCC	
SecI	CCNNGG	CCNNGG	
SecII	CCGG	CCGG	
SecIII	CCTNAGG	CCTNAGG	
SelI	CGCG	CGCG	
SelAI	GGNCC	GGNCC	
SenPI	CCNGG	CCNGG	
M.SenPI	CCNGG	CCNGG	
SenPT16I	CGGCCG	CGGCCG	
SenPT14bI	CCGCGG	CCGCGG	
SenpCI	CCGCGG	CCGCGG	
M.SenpCI	CCGCGG	CCGCGG	
SepI	ATGCAT	ATGCAT	
SeqAI	?	?	
SexI	CTCGAG	CTCGAG	
SexII	?	?	
SexAI	ACCWGGT	ACCWGGT	MN.
SexBI	CCGCGG	CCGCGG	
SexCI	CCGCGG	CCGCGG	
SfaI	GGCC	GGCC	
SfaAI	GCGATCGC	GCGATCGC	
SfaGUI	CCGG	CCGG	
SfaNI	GCATC	GATGC	IN.
M.SfaNI	GCATC	GCATC	
SfcI	CTRYAG	CTRYAG	N.
M.SfcI	CTRYAG	CTRYAG	
SfeI	CTRYAG	CTRYAG	
M.SfeI	CTRYAG	CTRYAG	
SfiI	GGCCNNNNNGGCC	GGCCNNNNNGGCC	ACFGIJKMNOQRSUVX.
M.SfiI	GGCCNNNNNGGCC	GGCCNNNNNGGCC	
SflI	CTGCAG	CTGCAG	
SflHK1794I	CCWGG	CCWGG	
SflHK2374I	CCWGG	CCWGG	
SflHK2731I	CCWGG	CCWGG	
SflHK6873I	CCWGG	CCWGG	
SflHK7234I	CCWGG	CCWGG	
SflHK7462I	CCWGG	CCWGG	
SflHK8401I	CCWGG	CCWGG	
SflHK10695I	CCSGG	CCSGG	
SflHK10790I	CCWGG	CCWGG	
SflHK11086I	CCSGG	CCSGG	
SflHK11087I	CCSGG	CCSGG	
SflHK11572I	CCSGG	CCSGG	
SflHK115731I	CCSGG	CCSGG	
Sfl2aI	CCWGG	CCWGG	
M.Sfl2aI	CCWGG	CCWGG	
Sfl2bI	CCWGG	CCWGG	
SfnI	GGWCC	GGWCC	
SfoI	GGCGCC	GGCGCC	N.
M.SfoI	GGCGCC	GGCGCC	
SfrI	CCGCGG	CCGCGG	
Sfr274I	CTCGAG	CTCGAG	IV.
Sfr303I	CCGCGG	CCGCGG	IV.
Sfr382I	CCGCGG	CCGCGG	
SfuI	TTCGAA	TTCGAA	M.
Sfu1762I	CTCGAG	CTCGAG	

SgaI	CTCGAG	CTCGAG	
SgfI	GCGATCGC	GCGATCGC	R.
Sgh1835I	GGWCC	GGWCC	
SgiI	CTGCAG	CTGCAG	
M.SglORF2102a	?	?	
SgoI	CTCGAG	CTCGAG	
SgrI	?	?	
Sgr20I	CCWGG	CCWGG	
Sgr1839I	TTCGAA	TTCGAA	
Sgr1841I	CTCGAG	CTCGAG	
SgrAI	CRCCGGYG	CRCCGGYG	MN.
M.SgrAI	CRCCGGYG	CRCCGGYG	
SgrBI	CCGCGG	CCGCGG	C.
SgrDI	CGTCGACG	CGTCGACG	
SgsI	GGCGCGCC	GGCGCGCC	F.
ShaI	GGGTC	GACCC	
ShyI	CCGCGG	CCGCGG	
Shy1766I	CTCGAG	CTCGAG	
ShyTI	?	?	
SimI	GGGTC	GACCC	
SinI	GGWCC	GGWCC	GR.
M.SinI	GGWCC	GGWCC	
SinAI	GGWCC	GGWCC	
SinBI	GGWCC	GGWCC	
SinCI	GGWCC	GGWCC	
SinDI	GGWCC	GGWCC	
SinEI	GGWCC	GGWCC	
SinFI	GGWCC	GGWCC	
SinGI	GGWCC	GGWCC	
SinHI	GGWCC	GGWCC	
SinJI	GGWCC	GGWCC	
SinMI	GATC	GATC	
SinMII	?	?	
SisI	?	?	
SkaI	GCCGGC	GCCGGC	
SkaII	CTGCAG	CTGCAG	
SlaI	CTCGAG	CTCGAG	C.
SlbI	GGTCTC	GAGACC	
SleI	CCWGG	CCWGG	
SliI	?	?	
SliII	?	?	
SluI	CTCGAG	CTCGAG	
Slu1777I	GCCGGC	GCCGGC	
SmaI	CCCGGG	CCCGGG	ABCFGHIJKMNOQRSUVXY.
M.SmaI	CCCGGG	CCCGGG	
M.SmaII	GATC	GATC	
SmaAI	CGTACG	CGTACG	
SmaAII	GACNNNGTC	GACNNNGTC	
SmaAIII	CGATCG	CGATCG	
SmaAIV	CAGCTG	CAGCTG	
M.SmeI	GANTC	GANTC	
SmiI	ATTTAAAT	ATTTAAAT	FIV.
SmiMI	CAYNNNNRTG	CAYNNNNRTG	I.
SmiMII	GATATC	GATATC	
SmiMBI	GATC	GATC	
SmlI	CTYRAG	CTYRAG	N.
SmoI	CTYRAG	CTYRAG	F.
Smo40529I	GCCGGC	GCCGGC	
SmuI	CCCGC	GCGGG	F.
SmuCI	ATGCAT	ATGCAT	
SmuEI	GGWCC	GGWCC	
SnaI	GTATAC	GTATAC	
Sna3286I	TCGCGA	TCGCGA	
SnaBI	TACGTA	TACGTA	ACKMNR.
M.SnaBI	TACGTA	TACGTA	
SniI	CCWGG	CCWGG	
SnoI	GTGCAC	GTGCAC	
SodI	?	?	
SodII	?	?	
SolI	GGATCC	GGATCC	
Sol3335I	CAGCTG	CAGCTG	
Sol10179I	CTCGAG	CTCGAG	
SpaI	CTCGAG	CTCGAG	
SpaHI	GCATGC	GCATGC	
SpaPI	GACNNNGTC	GACNNNGTC	
SpaPII	CGATCG	CGATCG	
SpaPIII	CAGCTG	CAGCTG	
SpaPIV	AAGCTT	AAGCTT	
SpaXI	GCATGC	GCATGC	
SpeI	ACTAGT	ACTAGT	ABGHJKMNOQRSUX.

M.SpeI	ACTAGT	ACTAGT	
SphI	GCATGC	GCATGC	ABCGHIJKMNOQRSVX.
M.SphI	GCATGC	GCATGC	
Sph1719I	CTCGAG	CTCGAG	
SplI	CGTACG	CGTACG	
SplII	GACNNNGTC	GACNNNGTC	
SplIII	GGCC	GGCC	
SplAI	CGTACG	CGTACG	
SplAII	GACNNNGTC	GACNNNGTC	
SplAIII	CGATCG	CGATCG	
SplAIV	CAGCTG	CAGCTG	
SpmI	ATCGAT	ATCGAT	
M.Spn6BI	TCTAGA	TCTAGA	
SpoI	TCGCGA	TCGCGA	
I-SpomI	GTGGTTGGACGGTATATCCACCACT	AGTGGTGGATATACCGTCCAACCAC	
M.SpomI	CCWGG	CCWGG	
SprLI	CTGCAG	CTGCAG	
M.SptAI	CAGCTG	CAGCTG	
SpuI	CCGCGG	CCGCGG	
SpvI	GGATCC	GGATCC	
SrfI	GCCCGGGC	GCCCGGGC	EO.
SriI	CTGCAG	CTGCAG	
SrifpI	CTCGAG	CTCGAG	
SrlI	GCCGGC	GCCGGC	
SrlII	ATGCAT	ATGCAT	
Srl19I	TTTAAA	TTTAAA	
Srl1DI	CTGCAG	CTGCAG	
Srl2DI	CTGCAG	CTGCAG	
Srl5DI	CTGCAG	CTGCAG	
Srl8DI	ATTAAT	ATTAAT	
Srl17DI	ATTAAT	ATTAAT	
Srl32DI	CTGCAG	CTGCAG	
Srl32DII	GAATTC	GAATTC	
Srl55DI	GAATTC	GAATTC	
Srl55DII	ATTAAT	ATTAAT	
Srl56DI	CTRYAG	CTRYAG	
Srl61DI	TTTAAA	TTTAAA	
Srl65DI	ATTAAT	ATTAAT	
Srl76DI	TTTAAA	TTTAAA	
Srl77DI	GCCGGC	GCCGGC	
Srr17I	ATTAAT	ATTAAT	
SruI	TTTAAA	TTTAAA	
Sru4DI	ATTAAT	ATTAAT	
Sru30DI	AGGCCT	AGGCCT	
SsaI	?	?	
SsbI	AAGCTT	AAGCTT	
SscI	?	?	
SscL1I	GANTC	GANTC	
M.SscL1I	GANTC	GANTC	
SseI	TGATCA	TGATCA	
SseII	CCGCGG	CCGCGG	
Sse9I	AATT	AATT	IV.
M.Sse9I	AATT	AATT	
Sse232I	CGCCGGCG	CGCCGGCG	
Sse1825I	GGGWCCC	GGGWCCC	
Sse8387I	CCTGCAGG	CCTGCAGG	AK.
Sse8647I	AGGWCCT	AGGWCCT	
SseAI	GGCGCC	GGCGCC	
SseBI	AGGCCT	AGGCCT	C.
SshAI	CCTNAGG	CCTNAGG	
SsiI	CCGC	GCGG	F.
SsiAI	GATC	GATC	
SsiBI	GATC	GATC	
SslI	CCWGG	CCWGG	
M.Ssl1I	GANTC	GANTC	
Ssl16215I	?	?	
Ssl16216I	?	?	
Ssl16217I	?	?	
Ssl16218I	?	?	
Ssl16219I	?	?	
SsmI	CTGATG	CATCAG	
SsmII	CCGCGG	CCGCGG	
SsoI	GAATTC	GAATTC	
M.SsoI	GAATTC	GAATTC	
SsoII	CCNGG	CCNGG	
M.SsoII	CCNGG	CCNGG	
M.SsoIII	?	?	
M.SsoIV	?	?	
M.SsoV	?	?	
SspI	AATATT	AATATT	ABCFGIJKMNOQRSUVX.

M.SspI	AATATT	AATATT	
Ssp1I	TTCGAA	TTCGAA	
Ssp2I	CCSGG	CCSGG	
Ssp4I	CTCGAG	CTCGAG	
Ssp12I	CTGCAG	CTGCAG	
Ssp14I	TTCGAA	TTCGAA	
Ssp27I	?	?	
Ssp34I	TTCGAA	TTCGAA	
Ssp42I	TTCGAA	TTCGAA	
Ssp43I	TTCGAA	TTCGAA	
Ssp45I	TTCGAA	TTCGAA	
Ssp47I	TTCGAA	TTCGAA	
Ssp48I	TTCGAA	TTCGAA	
Ssp152I	TTCGAA	TTCGAA	
Ssp1725I	CCGCGG	CCGCGG	
Ssp4800I	TGTACA	TGTACA	
Ssp5230I	GACGTC	GACGTC	
I-Ssp6803I	GTCGGGCTCATAACCCGAA	TTCGGGTTATGAGCCCGAC	
M.Ssp6803I	CGATCG	CGATCG	
Ssp27144I	ATCGAT	ATCGAT	
SspAI	CCWGG	CCWGG	
SspBI	TGTACA	TGTACA	M.
SspCI	GCCGGC	GCCGGC	
SspD5I	GGTGA	TCACC	
SspD5II	ATGCAT	ATGCAT	
SspJI	TACGTA	TACGTA	
SspJII	GRCGYC	GRCGYC	
SspKI	CGTACG	CGTACG	
SspM1I	TACGTA	TACGTA	
SspM1II	GRCGYC	GRCGYC	
SspM1III	GGYRCC	GGYRCC	
SspM2I	TACGTA	TACGTA	
SspM2II	GRCGYC	GRCGYC	
SspRFI	TTCGAA	TTCGAA	
SspXI	?	?	
SsrI	GTTAAC	GTTAAC	
M.SssI	CG	CG	N.
SstI	GAGCTC	GAGCTC	BC.
M.SstI	GAGCTC	GAGCTC	
SstII	CCGCGG	CCGCGG	B.
SstIII	?	?	
SstIV	TGATCA	TGATCA	
Sst12I	CTGCAG	CTGCAG	
Ssu211I	GATC	GATC	
M.Ssu211I	GATC	GATC	
Ssu212I	GATC	GATC	
M.Ssu212I	GATC	GATC	
Ssu220I	GATC	GATC	
M1.Ssu2479I	GATC	GATC	
M2.Ssu2479I	GATC	GATC	
R1.Ssu2479I	GATC	GATC	
R2.Ssu2479I	GATC	GATC	
M1.Ssu4109I	GATC	GATC	
M2.Ssu4109I	GATC	GATC	
R1.Ssu4109I	GATC	GATC	
R2.Ssu4109I	GATC	GATC	
M1.Ssu4961I	GATC	GATC	
M2.Ssu4961I	GATC	GATC	
R1.Ssu4961I	GATC	GATC	
R2.Ssu4961I	GATC	GATC	
M1.Ssu8074I	GATC	GATC	
M2.Ssu8074I	GATC	GATC	
R1.Ssu8074I	GATC	GATC	
R2.Ssu8074I	GATC	GATC	
M1.Ssu11318I	GATC	GATC	
M2.Ssu11318I	GATC	GATC	
R1.Ssu11318I	GATC	GATC	
R2.Ssu11318I	GATC	GATC	
M1.SsuDAT1I	GATC	GATC	
M2.SsuDAT1I	GATC	GATC	
R1.SsuDAT1I	GATC	GATC	
R2.SsuDAT1I	GATC	GATC	
SsuRBI	GATC	GATC	
SsvI	AGGCCT	AGGCCT	
StaI	CCGCGG	CCGCGG	
StaAI	CTCGAG	CTCGAG	
SteI	AGGCCT	AGGCCT	
SthI	GGTACC	GGTACC	
Sth117I	CCWGG	CCWGG	
Sth132I	CCCG	CGGG	

Sth134I	CCGG	CCGG	
Sth302I	CCWGG	CCWGG	
Sth302II	CCGG	CCGG	
Sth368I	GATC	GATC	
M.Sth368I	GATC	GATC	
Sth455I	CCWGG	CCWGG	
Sth4134I	?	?	
SthAI	GGTACC	GGTACC	
SthBI	GGTACC	GGTACC	
SthCI	GGTACC	GGTACC	
SthDI	GGTACC	GGTACC	
SthEI	GGTACC	GGTACC	
SthFI	GGTACC	GGTACC	
SthGI	GGTACC	GGTACC	
SthHI	GGTACC	GGTACC	
SthJI	GGTACC	GGTACC	
SthKI	GGTACC	GGTACC	
SthLI	GGTACC	GGTACC	
SthMI	GGTACC	GGTACC	
SthNI	GGTACC	GGTACC	
StmI	?	?	
StrI	CTCGAG	CTCGAG	U.
StsI	GGATG	CATCC	
M.StsI	GGATG	GGATG	
StuI	AGGCCT	AGGCCT	ABJKMNQRSUX.
M.StuI	AGGCCT	AGGCCT	
StyI	CCWWGG	CCWWGG	CJMNRS.
M.StyI	CCWWGG	CCWWGG	
StyD4I	CCNGG	CCNGG	N.
M.StyD4I	CCNGG	CCNGG	
M.StyDam	GATC	GATC	
M.Sty1344Dam	GATC	GATC	
M.Sty14028Dam	GATC	GATC	
StyLTI	CAGAG	CTCTG	
M.StyLTI	CAGAG	CAGAG	
StyLTII	?	?	
M.StyLTII	?	?	
StyLTIII	GAGNNNNNNRTAYG	CRTAYNNNNNNCTC	
M.StyLTIII	GAGNNNNNNRTAYG	GAGNNNNNNRTAYG	
M.StyLT2Dam	GATC	GATC	
StySBLI	CGANNNNNNTACC	GGTANNNNNNTCG	
M.StySBLI	CGANNNNNNTACC	CGANNNNNNTACC	
StySEAI	ACANNNNNNTYCA	TGRANNNNNNTGT	
M.StySEAI	ACANNNNNNTYCA	ACANNNNNNTYCA	
StySENI	CGANNNNNNTACC	GGTANNNNNNTCG	
M.StySENI	CGANNNNNNTACC	CGANNNNNNTACC	
StySGI	TAANNNNNNRTCG	CGAYNNNNNNTTA	
M.StySGI	TAANNNNNNRTCG	TAANNNNNNRTCG	
StySJI	GAGNNNNNNGTRC	GYACNNNNNNCTC	
M.StySJI	GAGNNNNNNGTRC	GAGNNNNNNGTRC	
StySKI	CGATNNNNNNNGTTA	TAACNNNNNNNATCG	
M.StySKI	CGATNNNNNNNGTTA	CGATNNNNNNNGTTA	
StySPI	AACNNNNNNGTRC	GYACNNNNNNGTT	
M.StySPI	AACNNNNNNGTRC	AACNNNNNNGTRC	
StySQI	AACNNNNNNRTAYG	CRTAYNNNNNNGTT	
M.StySQI	AACNNNNNNRTAYG	AACNNNNNNRTAYG	
StySTI	?	?	
SuaI	GGCC	GGCC	
M.SuaI	GGCC	GGCC	
SulI	GGCC	GGCC	
SunI	CGTACG	CGTACG	
SurI	GGATCC	GGATCC	
F-SuvI	?	?	
Sve194I	CTCGAG	CTCGAG	
SviI	TTCGAA	TTCGAA	
SwaI	ATTTAAAT	ATTTAAAT	GKMNS.
M.SwaI	ATTTAAAT	ATTTAAAT	
SynI	GGWCC	GGWCC	
SynII	GAANNNNTTC	GAANNNNTTC	
TaaI	ACNGT	ACNGT	F.
M.TaeI	?	?	
M.TaeII	TGATCA	TGATCA	
M.TaeCDnmtI	?	?	
TaiI	ACGT	ACGT	F.
TaqI	TCGA	TCGA	ABCFGIJKMNOQRSUVXY.
M.TaqI	TCGA	TCGA	N.
TaqII	GACCGA	TCGGTC	VX.
TaqII	CACCCA	TGGGTG	VX.
Taq20I	TCGA	TCGA	
Taq52I	GCWGC	GCWGC	

TaqXI	CCWGG	CCWGG	
TasI	AATT	AATT	F.
TatI	WGTACW	WGTACW	F.
TauI	GCSGC	GCSGC	F.
TauII	CGGCCG	CGGCCG	
Tbr51I	TCGA	TCGA	
TceI	GAAGA	TCTTC	
TdeI	GATC	GATC	
TdeII	CTCTTC	GAAGAG	
M.TdeII	CTCTTC	CTCTTC	
TdeIII	GGNCC	GGNCC	
M.TdeIII	GGNCC	GGNCC	
TelI	GACNNNGTC	GACNNNGTC	
F-TevI	GAAACACAAGAAATGTTTAGTAAA	TTTACTAAACATTTCTTGTGTTTC	
I-TevI	AGTGGTATCAACGCTCAGTAGATG	CATCTACTGAGCGTTGATACCACT	
F-TevII	TTTAATCCTCGCTTCAGATATGGCAACTG	CAGTTGCCATATCTGAAGCGAGGATTAAA	
I-TevII	GCTTATGAGTATGAAGTGAACACGTTATTC	GAATAACGTGTTCACTTCATACTCATAAGC	
F-TevIII	AGAAGAACATGTGGTATTG	CAATACCACATGTTCTTCT	
I-TevIII	TATGTATCTTTTGCGTGTACCTTTAACTTC	GAAGTTAAAGGTACACGCAAAAGATACATA	
TfeI	?	?	
TfiI	GAWTC	GAWTC	N.
M.TfiI	GAWTC	GAWTC	
TfiA3I	TCGA	TCGA	
TfiTok4A2I	TCGA	TCGA	
TfiTok6A1I	TCGA	TCGA	
M.TfiTok6A1I	TCGA	TCGA	
TflI	TCGA	TCGA	
PI-TfuI	TAGATTTTAGGTCGCTATATCCTTCC	GGAAGGATATAGCGACCTAAAATCTA	
PI-TfuII	TAYGCNGAYACNGACGGYTTYT	ARAARCCGTCNGTRTCNGCRTA	
TglI	CCGCGG	CCGCGG	
ThaI	CGCG	CGCG	
M.ThaI	CGCG	CGCG	
M.ThaII	GATC	GATC	
M.ThaIII	GANTC	GANTC	
PI-ThyI	TAYGCNGAYACNGACGGYTTYT	ARAARCCGTCNGTRTCNGCRTA	
TliI	CTCGAG	CTCGAG	N.
M.TliI	CTCGAG	CTCGAG	
PI-TliI	TAYGCNGAYACNGACGGYTTYT	ARAARCCGTCNGTRTCNGCRTA	
PI-TliII	AAATTGCTTGCAAACAGCTATTACGGCTAT	ATAGCCGTAATAGCTGTTTGCAAGCAATTT	
TmaI	CGCG	CGCG	
M.TmaI	CGCG	CGCG	
TmiI	?	?	
Tmu1I	CCSGG	CCSGG	
TnoI	?	?	
M.TpaI	GATC	GATC	
TrsKTI	GATC	GATC	
M.TrsKTI	GATC	GATC	
TrsKTII	GACNNNGTC	GACNNNGTC	
TrsKTIII	CATATG	CATATG	
TrsSI	GATC	GATC	
M.TrsSI	GATC	GATC	
TrsSII	GACNNNNNNGTC	GACNNNNNNGTC	
TrsTI	GATC	GATC	
M.TrsTI	GATC	GATC	
TrsTII	CTTAAG	CTTAAG	
TruI	GGWCC	GGWCC	
TruII	GATC	GATC	
Tru1I	TTAA	TTAA	F.
Tru9I	TTAA	TTAA	GIMRV.
Tru28I	GGWCC	GGWCC	
Tru201I	RGATCY	RGATCY	
TscI	ACGT	ACGT	
TscHI	?	?	
Tsc4aI	TCGA	TCGA	
TseI	GCWGC	GCWGC	N.
M.TseI	GCWGC	GCWGC	
TseAI	GDGCHC	GDGCHC	
TseBI	GCWGC	GCWGC	
TseCI	AATT	AATT	
TseDI	RCCGGY	RCCGGY	
TsoI	TARCCA	TGGYTA	F.
TspI	GACNNNGTC	GACNNNGTC	
Tsp1I	ACTGG	CCAGT	
Tsp32I	TCGA	TCGA	
M.Tsp32I	TCGA	TCGA	
Tsp32II	TCGA	TCGA	
Tsp45I	GTSAC	GTSAC	N.
M.Tsp45I	GTSAC	GTSAC	
Tsp49I	ACGT	ACGT	
I-Tsp061I	CTTCAGTATGCCCCGAAAC	GTTTCGGGGCATACTGAAG	

Tsp132I	GGCC	GGCC	
Tsp133I	GATC	GATC	
Tsp219I	GCCNNNNNGGC	GCCNNNNNGGC	
Tsp266I	GGCC	GGCC	
Tsp273I	GATATC	GATATC	
Tsp273II	GGCC	GGCC	
Tsp281I	GGCC	GGCC	
Tsp301I	GGWCC	GGWCC	
Tsp358I	TCGA	TCGA	
Tsp504I	CGGCCG	CGGCCG	
Tsp505I	TCGA	TCGA	
Tsp507I	TCCGGA	TCCGGA	
Tsp509I	AATT	AATT	N.
M.Tsp509I	AATT	AATT	
Tsp510I	TCGA	TCGA	
Tsp514I	TCCGGA	TCCGGA	
Tsp560I	GGCC	GGCC	
TspAI	CCWGG	CCWGG	
TspAK13D21I	TCGA	TCGA	
TspAK16D24I	TCGA	TCGA	
TspBI	CCRYGG	CCRYGG	
Tsp4CI	ACNGT	ACNGT	
TspDTI	ATGAA	TTCAT	VX.
TspEI	AATT	AATT	O.
Tsp8EI	GCCNNNNNGGC	GCCNNNNNGGC	
TspGWI	ACGGA	TCCGT	VX.
TspGWII	CTGCAG	CTGCAG	
TspIDSI	ACGT	ACGT	
TspMI	CCCGGG	CCCGGG	N.
TspNI	TCGA	TCGA	
TspRI	CASTG	CASTG	GN.
M.TspRI	CASTG	CASTG	
TspVi4AI	TCGA	TCGA	
TspVil3I	TCGA	TCGA	
TspWAM8AI	ACGT	ACGT	
TspZNI	GGCC	GGCC	
TssI	GAGNNNCTC	GAGNNNCTC	
TstI	CACNNNNNNTCC	GGANNNNNNGTG	F.
TstI	GGANNNNNNGTG	CACNNNNNNTCC	F.
TsuI	GCGAC	GTCGC	
TteI	GACNNNGTC	GACNNNGTC	
TteAI	GGCC	GGCC	
Tth24I	TCGA	TCGA	
Tth111I	GACNNNGTC	GACNNNGTC	GIKNQRVX.
M.Tth111I	GACNNNGTC	GACNNNGTC	
Tth111II	CAARCA	TGYTTG	
M.TthBI	?	?	
TthHB8I	TCGA	TCGA	
M.TthHB8I	TCGA	TCGA	
TthHB27I	CAARCA	TGYTTG	
TthRQI	TCGA	TCGA	
TtmI	ACGT	ACGT	
TtmII	GCGCGC	GCGCGC	
TtnI	GGCC	GGCC	
TtoI	CCGCGG	CCGCGG	
TtrI	GACNNNGTC	GACNNNGTC	
TveI	?	?	
M.TvoDam	GATC	GATC	
I-TwoI	TCTTGCACCTACACAATCCA	TGGATTGTGTAGGTGCAAGA	
Uba4I	GATC	GATC	
Uba6I	ACGCGT	ACGCGT	
Uba9I	GGCC	GGCC	
Uba11I	CCWGG	CCWGG	
Uba13I	CCWGG	CCWGG	
Uba17I	CCNGG	CCNGG	
Uba19I	GGATCC	GGATCC	
Uba20I	CCWGG	CCWGG	
Uba22I	ATCGAT	ATCGAT	
Uba24I	ATCGAT	ATCGAT	
Uba30I	ATCGAT	ATCGAT	
Uba31I	GGATCC	GGATCC	
Uba34I	ATCGAT	ATCGAT	
Uba36I	YGGCCR	YGGCCR	
Uba38I	GGATCC	GGATCC	
Uba39I	GRGCYC	GRGCYC	
Uba40I	AGGCCT	AGGCCT	
Uba41I	CCSGG	CCSGG	
Uba42I	CCSGG	CCSGG	
Uba43I	ATCGAT	ATCGAT	
Uba46I	CTGCAG	CTGCAG	

Uba48I	GGWCC	GGWCC
Uba51I	GGATCC	GGATCC
Uba54I	GGCC	GGCC
Uba57I	GRGCYC	GRGCYC
Uba58I	GAATTC	GAATTC
Uba59I	GATC	GATC
Uba61I	GGCC	GGCC
Uba62I	GGWCC	GGWCC
Uba65I	GGTCTC	GAGACC
Uba66I	CCGCGG	CCGCGG
Uba69I	GCGCGC	GCGCGC
Uba71I	CTGCAG	CTGCAG
Uba72I	CTGCAG	CTGCAG
Uba76I	GGTACC	GGTACC
Uba77I	CCGCGG	CCGCGG
Uba81I	CCWGG	CCWGG
Uba82I	CCWGG	CCWGG
Uba83I	AAGCTT	AAGCTT
Uba84I	GGTCTC	GAGACC
Uba85I	GGTACC	GGTACC
Uba86I	GGTACC	GGTACC
Uba87I	GGTACC	GGTACC
Uba88I	GGATCC	GGATCC
Uba89I	GTCGAC	GTCGAC
Uba90I	CCGCGG	CCGCGG
Uba1093I	CCGCGG	CCGCGG
Uba1094I	AGTACT	AGTACT
Uba1095I	CCGCGG	CCGCGG
Uba1096I	ATCGAT	ATCGAT
Uba1097I	GGCC	GGCC
Uba1098I	GGATCC	GGATCC
Uba1099I	GGNCC	GGNCC
Uba1100I	ATCGAT	ATCGAT
Uba1101I	GATC	GATC
Uba1111I	CCGCGG	CCGCGG
Uba1112I	CTGCAG	CTGCAG
Uba1113I	CCGCGG	CCGCGG
Uba1114I	CCWGG	CCWGG
Uba1115I	CTGCAG	CTGCAG
Uba1116I	CTGCAG	CTGCAG
Uba1117I	TCGCGA	TCGCGA
Uba1118I	CCWGG	CCWGG
Uba1119I	CTGCAG	CTGCAG
Uba1120I	CCWGG	CCWGG
Uba1121I	CCWGG	CCWGG
Uba1122I	GCCGGC	GCCGGC
Uba1123I	CTGCAG	CTGCAG
Uba1124I	GRGCYC	GRGCYC
Uba1125I	CCWGG	CCWGG
Uba1126I	CCGCGG	CCGCGG
Uba1127I	GGYRCC	GGYRCC
Uba1128I	CCGG	CCGG
Uba1129I	CGATCG	CGATCG
Uba1130I	CTCGAG	CTCGAG
Uba1131I	GGWCC	GGWCC
Uba1133I	ATCGAT	ATCGAT
Uba1134I	GGNCC	GGNCC
Uba1136I	TCCGGA	TCCGGA
Uba1137I	ATCGAT	ATCGAT
Uba1138I	ATCGAT	ATCGAT
Uba1139I	CGATCG	CGATCG
Uba1140I	GGCC	GGCC
Uba1141I	CCGG	CCGG
Uba1142I	GRGCYC	GRGCYC
Uba1144I	ATCGAT	ATCGAT
Uba1145I	ATCGAT	ATCGAT
Uba1146I	GGCC	GGCC
Uba1147I	GGCC	GGCC
Uba1148I	CTCGAG	CTCGAG
Uba1149I	CTGCAG	CTGCAG
Uba1150I	GGCC	GGCC
Uba1152I	GGCC	GGCC
Uba1153I	GGCC	GGCC
Uba1154I	CTCGAG	CTCGAG
Uba1155I	GGCC	GGCC
Uba1156I	GGGCCC	GGGCCC
Uba1157I	GGGCCC	GGGCCC
Uba1158I	AGTACT	AGTACT
Uba1159I	GRGCYC	GRGCYC
Uba1160I	GGNCC	GGNCC

Uba1161I	ATCGAT	ATCGAT
Uba1162I	GCATGC	GCATGC
Uba1163I	GGATCC	GGATCC
Uba1164I	GGNCC	GGNCC
Uba1164II	AAGCTT	AAGCTT
Uba1165I	GGGCCC	GGGCCC
Uba1166I	CTCGAG	CTCGAG
Uba1167I	GGATCC	GGATCC
Uba1168I	ATCGAT	ATCGAT
Uba1169I	GGCC	GGCC
Uba1170I	AGGCCT	AGGCCT
Uba1171I	CCWGG	CCWGG
Uba1172I	GGATCC	GGATCC
Uba1173I	GGATCC	GGATCC
Uba1174I	GGCC	GGCC
Uba1175I	GGCC	GGCC
Uba1176I	GGCC	GGCC
Uba1177I	GATC	GATC
Uba1178I	GGCC	GGCC
Uba1179I	GGCC	GGCC
Uba1180I	AGGCCT	AGGCCT
Uba1181I	CCWGG	CCWGG
Uba1182I	GATC	GATC
Uba1183I	GATC	GATC
Uba1184I	CTGCAG	CTGCAG
Uba1184II	CCTNAGG	CCTNAGG
Uba1185I	CCWGG	CCWGG
Uba1186I	CTGCAG	CTGCAG
Uba1187I	CCGCGG	CCGCGG
Uba1188I	YGGCCR	YGGCCR
Uba1189I	CCWGG	CCWGG
Uba1190I	GACNNNNNGTC	GACNNNNNGTC
Uba1191I	GACNNNNNGTC	GACNNNNNGTC
Uba1192I	CTCTTC	GAAGAG
Uba1193I	CCWGG	CCWGG
Uba1195I	ATCGAT	ATCGAT
Uba1196I	ATCGAT	ATCGAT
Uba1197I	ATCGAT	ATCGAT
Uba1198I	ATCGAT	ATCGAT
Uba1199I	ATCGAT	ATCGAT
Uba1200I	ATCGAT	ATCGAT
Uba1201I	GGTACC	GGTACC
Uba1202I	GGGCCC	GGGCCC
Uba1203I	GTGCAC	GTGCAC
Uba1204I	GATC	GATC
Uba1205I	GGATCC	GGATCC
Uba1205II	CYCGRG	CYCGRG
Uba1206I	GRGCYC	GRGCYC
Uba1207I	GGCC	GGCC
Uba1208I	GGCC	GGCC
Uba1209I	GGCC	GGCC
Uba1210I	GGCC	GGCC
Uba1211I	CTGCAG	CTGCAG
Uba1212I	CTGCAG	CTGCAG
Uba1213I	CTGCAG	CTGCAG
Uba1214I	GGCC	GGCC
Uba1215I	CTGCAG	CTGCAG
Uba1216I	CTGCAG	CTGCAG
Uba1217I	AGGCCT	AGGCCT
Uba1218I	CCWGG	CCWGG
Uba1219I	AAGCTT	AAGCTT
Uba1220I	CCCGGG	CCCGGG
Uba1221I	GCTNAGC	GCTNAGC
Uba1222I	GCTNAGC	GCTNAGC
Uba1223I	GGCC	GGCC
Uba1224I	GGATCC	GGATCC
Uba1225I	CTGCAG	CTGCAG
Uba1226I	GCATGC	GCATGC
Uba1227I	CAGCTG	CAGCTG
Uba1228I	GGCC	GGCC
Uba1229I	CCGCGG	CCGCGG
Uba1230I	GGCC	GGCC
Uba1231I	GGCC	GGCC
Uba1232I	CTGCAG	CTGCAG
Uba1233I	ATCGAT	ATCGAT
Uba1234I	CCGCGG	CCGCGG
Uba1235I	GGCC	GGCC
Uba1237I	CTCGAG	CTCGAG
Uba1238I	ATCGAT	ATCGAT
Uba1239I	AGGCCT	AGGCCT

Uba1240I	TACGTA	TACGTA
Uba1241I	GGGCCC	GGGCCC
Uba1242I	GGATCC	GGATCC
Uba1243I	CCWGG	CCWGG
Uba1244I	CCGCGG	CCGCGG
Uba1245I	CAGCTG	CAGCTG
Uba1246I	ATCGAT	ATCGAT
Uba1248I	CTCGAG	CTCGAG
Uba1249I	GGWCC	GGWCC
Uba1250I	GGATCC	GGATCC
Uba1256I	CTGCAG	CTGCAG
Uba1257I	ATCGAT	ATCGAT
Uba1258I	GGATCC	GGATCC
Uba1259I	GATC	GATC
Uba1262I	CTGCAG	CTGCAG
Uba1263I	GRGCYC	GRGCYC
Uba1264I	GRGCYC	GRGCYC
Uba1266I	CTTAAG	CTTAAG
Uba1267I	CCGG	CCGG
Uba1271I	CTCGAG	CTCGAG
Uba1272I	GGWCC	GGWCC
Uba1275I	ATCGAT	ATCGAT
Uba1276I	CTCTTC	GAAGAG
Uba1278I	GGWCC	GGWCC
Uba1279I	TCCGGA	TCCGGA
Uba1280I	CCSGG	CCSGG
Uba1282I	TGATCA	TGATCA
Uba1283I	TGATCA	TGATCA
Uba1284I	GCTNAGC	GCTNAGC
Uba1286I	ATCGAT	ATCGAT
Uba1287I	CTGCAG	CTGCAG
Uba1288I	GGCC	GGCC
Uba1289I	CCTNNNNNAGG	CCTNNNNNAGG
Uba1290I	CCTNNNNNAGG	CCTNNNNNAGG
Uba1291I	GGTNACC	GGTNACC
Uba1292I	GGCC	GGCC
Uba1293I	GGCC	GGCC
Uba1294I	CCTNAGG	CCTNAGG
Uba1294II	CTGCAG	CTGCAG
Uba1295I	ATCGAT	ATCGAT
Uba1296I	CTGCAG	CTGCAG
Uba1297I	GGATCC	GGATCC
Uba1298I	CTCGAG	CTCGAG
Uba1299I	CTTAAG	CTTAAG
Uba1302I	GGATCC	GGATCC
Uba1303I	CGRYCG	CGRYCG
Uba1304I	GGWCC	GGWCC
Uba1305I	GGNNCC	GGNNCC
Uba1306I	CCGCGG	CCGCGG
Uba1307I	GRGCYC	GRGCYC
Uba1308I	CCTNNNNNAGG	CCTNNNNNAGG
Uba1309I	CCTNNNNNAGG	CCTNNNNNAGG
Uba1310I	CCTNNNNNAGG	CCTNNNNNAGG
Uba1311I	CCWWGG	CCWWGG
Uba1312I	CTTAAG	CTTAAG
Uba1313I	CTTAAG	CTTAAG
Uba1314I	GGWCC	GGWCC
Uba1315I	ATCGAT	ATCGAT
Uba1316I	GGTCTC	GAGACC
Uba1317I	GATC	GATC
Uba1318I	CCSGG	CCSGG
Uba1319I	GGCC	GGCC
Uba1320I	GCTNAGC	GCTNAGC
Uba1321I	CGCG	CGCG
Uba1322I	GGCC	GGCC
Uba1323I	GATC	GATC
Uba1324I	GGATCC	GGATCC
Uba1325I	GGATCC	GGATCC
Uba1326I	RGGNCCY	RGGNCCY
Uba1327I	YGGCCR	YGGCCR
Uba1328I	CTGCAG	CTGCAG
Uba1329I	GRGCYC	GRGCYC
Uba1330I	GRGCYC	GRGCYC
Uba1331I	CTTAAG	CTTAAG
Uba1332I	CCTNAGG	CCTNAGG
Uba1333I	CCTNAGG	CCTNAGG
Uba1334I	GGATCC	GGATCC
Uba1335I	CTCGAG	CTCGAG
Uba1336I	GGCC	GGCC
Uba1337I	CTGCAG	CTGCAG

Uba1338I	CCGG	CCGG
Uba1339I	GGATCC	GGATCC
Uba1342I	ATCGAT	ATCGAT
Uba1343I	GGTCTC	GAGACC
Uba1346I	GGATCC	GGATCC
Uba1347I	CCSGG	CCSGG
Uba1353I	ATGCAT	ATGCAT
Uba1355I	CCGG	CCGG
Uba1357I	GRGCYC	GRGCYC
Uba1362I	GDGCHC	GDGCHC
Uba1363I	GRGCYC	GRGCYC
Uba1364I	CCGCGG	CCGCGG
Uba1366I	GATC	GATC
Uba1366II	ATCGAT	ATCGAT
Uba1367I	ATGCAT	ATGCAT
Uba1368I	GGGCCC	GGGCCC
Uba1369I	CCGCGG	CCGCGG
Uba1370I	CCSGG	CCSGG
Uba1371I	AGGCCT	AGGCCT
Uba1372I	CCSGG	CCSGG
Uba1373I	GGWCC	GGWCC
Uba1374I	CTTAAG	CTTAAG
Uba1375I	TCCGGA	TCCGGA
Uba1376I	CCSGG	CCSGG
Uba1377I	GGCC	GGCC
Uba1378I	CCSGG	CCSGG
Uba1379I	ATCGAT	ATCGAT
Uba1380I	ATCGAT	ATCGAT
Uba1381I	GRCGYC	GRCGYC
Uba1382I	GAATGC	GCATTC
Uba1383I	GGATCC	GGATCC
Uba1384I	ATGCAT	ATGCAT
Uba1385I	TTCGAA	TTCGAA
Uba1386I	TCGCGA	TCGCGA
Uba1387I	GTGCAC	GTGCAC
Uba1388I	GGCC	GGCC
Uba1389I	CCSGG	CCSGG
Uba1391I	CCNGG	CCNGG
Uba1392I	GGCC	GGCC
Uba1393I	CCCGGG	CCCGGG
Uba1394I	ATCGAT	ATCGAT
Uba1395I	GGCC	GGCC
Uba1397I	CTCGAG	CTCGAG
Uba1398I	GGATCC	GGATCC
Uba1399I	CTGCAG	CTGCAG
Uba1400I	GATATC	GATATC
Uba1401I	CCSGG	CCSGG
Uba1402I	GGATCC	GGATCC
Uba1403I	AGGCCT	AGGCCT
Uba1404I	CGCG	CGCG
Uba1405I	CGCG	CGCG
Uba1408I	GGCC	GGCC
Uba1408II	GTTAAC	GTTAAC
Uba1409I	GRGCYC	GRGCYC
Uba1410I	CCWGG	CCWGG
Uba1411I	CTGCAG	CTGCAG
Uba1412I	ATCGAT	ATCGAT
Uba1413I	GGWCC	GGWCC
Uba1414I	GGATCC	GGATCC
Uba1415I	GAATGC	GCATTC
Uba1416I	ATCGAT	ATCGAT
Uba1417I	CTGCAG	CTGCAG
Uba1418I	GGCC	GGCC
Uba1419I	AGGCCT	AGGCCT
Uba1420I	CTTAAG	CTTAAG
Uba1421I	GRGCYC	GRGCYC
Uba1422I	GGCC	GGCC
Uba1423I	CCSGG	CCSGG
Uba1424I	CCSGG	CCSGG
Uba1425I	TCCGGA	TCCGGA
Uba1426I	CTTAAG	CTTAAG
Uba1427I	ATCGAT	ATCGAT
Uba1428I	CCWGG	CCWGG
Uba1429I	GGCC	GGCC
Uba1430I	ATCGAT	ATCGAT
Uba1431I	TGATCA	TGATCA
Uba1432I	RGATCY	RGATCY
Uba1433I	AGCT	AGCT
Uba1435I	AAGCTT	AAGCTT
Uba1436I	CYCGRG	CYCGRG

Uba1437I	CTGGAG	CTCCAG	
Uba1438I	GGWCC	GGWCC	
Uba1439I	CCGG	CCGG	
Uba1440I	CYCGRG	CYCGRG	
Uba1441I	AGCT	AGCT	
Uba1442I	CCNNGG	CCNNGG	
Uba1443I	CTTAAG	CTTAAG	
Uba1444I	CTGGAG	CTCCAG	
Uba1445I	GGNNCC	GGNNCC	
Uba1446I	CGCG	CGCG	
Uba1447I	TGATCA	TGATCA	
Uba1448I	CTCGAG	CTCGAG	
Uba1449I	GGCC	GGCC	
Uba1450I	GGCC	GGCC	
Uba1451I	ATCGAT	ATCGAT	
Uba1452I	TTCGAA	TTCGAA	
Uba1453I	ATCGAT	ATCGAT	
Uba4009I	GGATCC	GGATCC	
Uba153AI	CAGCTG	CAGCTG	
UbaF9I	TACNNNNNRTGT	ACAYNNNNNGTA	
UbaF11I	TCGTA	TACGA	
UbaHKAI	CCGCGG	CCGCGG	
UbaHKBI	CTGCAG	CTGCAG	
UbaM39I	CAGCTG	CAGCTG	
UbaPI	CGAACG	CGTTCG	
Umi5I	CYCGRG	CYCGRG	
Umi7I	TGATCA	TGATCA	
UnbI	GGNCC	GGNCC	
Uth549I	GGCC	GGCC	
Uth554I	GGWCC	GGWCC	
Uth555I	GGCC	GGCC	
Uth557I	GGCC	GGCC	
Uur960I	GCNGC	GCNGC	
VanI	GCCNNNNNGGC	GCCNNNNNGGC	
Van91I	CCANNNNNTGG	CCANNNNNTGG	AFGKM.
Van91II	GAATTC	GAATTC	
M.Van91II	GAATTC	GAATTC	
Van91III	GGCC	GGCC	
Van91IV	?	?	
M.Vch0395Dam	GATC	GATC	
M.VchK139I	GATC	GATC	
VchN100I	GAATTC	GAATTC	
VchO2I	GAATTC	GAATTC	
VchO6I	?	?	
VchO24I	?	?	
VchO25I	GTATAC	GTATAC	
VchO44I	AGGCCT	AGGCCT	
VchO49I	AGTACT	AGTACT	
VchO52I	?	?	
VchO60I	?	?	
VchO66I	GGNCC	GGNCC	
VchO68I	GCATGC	GCATGC	
VchO70I	TCGCGA	TCGCGA	
VchO85I	GGNCC	GGNCC	
VchO87I	CTGCAG	CTGCAG	
VchO90I	GGNCC	GGNCC	
VfiI	CTTAAG	CTTAAG	
VhaI	GGCC	GGCC	
Vha44I	GATC	GATC	
Vha464I	CTTAAG	CTTAAG	IV.
Vha1168I	GGCC	GGCC	
VneI	GTGCAC	GTGCAC	IV.
VneAI	RGGNCCY	RGGNCCY	
VniI	GGCC	GGCC	
VpaK11I	GGWCC	GGWCC	
VpaK15I	GGNCC	GGNCC	
VpaK25I	GGNCC	GGNCC	
VpaK32I	GCTCTTC	GAAGAGC	
VpaK57I	GGTCTC	GAGACC	
VpaK65I	GGWCC	GGWCC	
VpaK3AI	CACGTG	CACGTG	
VpaK4AI	CTGCAG	CTGCAG	
VpaK7AI	GGWCC	GGWCC	
VpaK8AI	?	?	
VpaK9AI	GGNCC	GGNCC	
VpaK11AI	GGWCC	GGWCC	
VpaK12AI	?	?	
VpaK13AI	GGWCC	GGWCC	
VpaK19AI	GGNCC	GGNCC	
VpaK29AI	CTGCAG	CTGCAG	

VpaK50AI	?	?	
VpaK55AI	?	?	
VpaK56AI	?	?	
VpaK57AI	GGTCTC	GAGACC	
VpaK3BI	CACGTG	CACGTG	
VpaK4BI	CTGCAG	CTGCAG	
VpaK11BI	GGWCC	GGWCC	K.
VpaK12BI	?	?	
VpaK19BI	GGNCC	GGNCC	
VpaK11CI	GGWCC	GGWCC	
VpaK11DI	GGWCC	GGWCC	
VpaKutAI	GGNCC	GGNCC	
VpaKutBI	GGNCC	GGNCC	
VpaKutCI	?	?	
VpaKutDI	?	?	
VpaKutEI	CTCTTC	GAAGAG	
VpaKutFI	CTCTTC	GAAGAG	
VpaKutGI	CTGCAG	CTGCAG	
VpaKutHI	GGTCTC	GAGACC	
VpaKutJI	GGNCC	GGNCC	
VpaO5I	CTCTTC	GAAGAG	
VspI	ATTAAT	ATTAAT	FIRV.
M.VspI	ATTAAT	ATTAAT	
Vsp2246I	GGYRCC	GGYRCC	
XagI	CCTNNNNNAGG	CCTNNNNNAGG	F.
XamI	GTCGAC	GTCGAC	
M.XamI	GTCGAC	GTCGAC	
XapI	RAATTY	RAATTY	F.
XbaI	TCTAGA	TCTAGA	ABCFGHIJKMNOQRSUVXY.
M.XbaI	TCTAGA	TCTAGA	
XcaI	GTATAC	GTATAC	
XceI	RCATGY	RCATGY	F.
XciI	GTCGAC	GTCGAC	
XcmI	CCANNNNNNNNNTGG	CCANNNNNNNNNTGG	N.
M.XcmI	CCANNNNNNNNNTGG	CCANNNNNNNNNTGG	
XcyI	CCCGGG	CCCGGG	
M.XcyI	CCCGGG	CCCGGG	
Xgl3216I	CGATCG	CGATCG	
Xgl3217I	CGATCG	CGATCG	
Xgl3218I	CGATCG	CGATCG	
Xgl3219I	CGATCG	CGATCG	
Xgl3220I	CGATCG	CGATCG	
XhoI	CTCGAG	CTCGAG	ABFGHJKMNOQRSUXY.
M.XhoI	CTCGAG	CTCGAG	
XhoII	RGATCY	RGATCY	GMR.
M.XhoII	RGATCY	RGATCY	
M.XlaDnmt1	?	?	
XmaI	CCCGGG	CCCGGG	INRUV.
M.XmaI	CCCGGG	CCCGGG	
XmaII	CTGCAG	CTGCAG	
XmaIII	CGGCCG	CGGCCG	
M.XmaIII	CGGCCG	CGGCCG	
XmaCI	CCCGGG	CCCGGG	M.
XmaJI	CCTAGG	CCTAGG	F.
M.XmaXhDnmt1	?	?	
XmiI	GTMKAC	GTMKAC	F.
XmlI	CGATCG	CGATCG	
XmlAI	CGATCG	CGATCG	
XmnI	GAANNNNTTC	GAANNNNTTC	GNRU.
M.XmnI	GAANNNNTTC	GAANNNNTTC	
XniI	CGATCG	CGATCG	
XorI	CTGCAG	CTGCAG	
XorII	CGATCG	CGATCG	
M.XorII	CGATCG	CGATCG	
XpaI	CTCGAG	CTCGAG	
XphI	CTGCAG	CTGCAG	
M.XphI	CTGCAG	CTGCAG	
XspI	CTAG	CTAG	K.
XveI	CTGCAG	CTGCAG	
M.XveI	CTGCAG	CTGCAG	
M.XveII	CCCGGG	CCCGGG	
YenI	CTGCAG	CTGCAG	
M.YenI	CTGCAG	CTGCAG	
YenAI	CTGCAG	CTGCAG	
YenBI	CTGCAG	CTGCAG	
YenCI	CTGCAG	CTGCAG	
YenDI	CTGCAG	CTGCAG	
YenEI	CTGCAG	CTGCAG	
M.YenSDam	GATC	GATC	
M.YenWI	CTGCAG	CTGCAG	

```
M.YpsADam     GATC                           GATC
M.YpsDam      GATC                           GATC
ZanI          CCWGG                          CCWGG
PI-ZbaI       TACGTTGGTTGTGGTGAAAGAGGAAAAGAG CTCTTTTCCTCTTTCACCACAACCAACGTA
ZhoI          ATCGAT                         ATCGAT
M.ZmaIIA      ?                              ?
M.ZmaV        ?                              ?
M.ZmaDRM1     ?                              ?
M.ZmaDnmt1    ?                              ?
ZraI          GACGTC                         GACGTC                         INV.
ZrmI          AGTACT                         AGTACT                         I.
Zsp2I         ATGCAT                         ATGCAT                         IV.
```

(*):

A=GE Healthcare (8/05)

B=Invitrogen Corporation (8/05)

C=Minotech Biotechnology (9/05)

E=Stratagene (9/05)

F=Fermentas International Inc. (2/06)

G=Qbiogene (9/05)

H=American Allied Biochemical, Inc. (9/05)

I=SibEnzyme Ltd. (2/06)

J=Nippon Gene Co., Ltd. (8/05)

K=Takara Bio Inc. (9/05)

M=Roche Applied Science (8/05)

N=New England Biolabs (4/06)

O=Toyobo Biochemicals (9/05)

Q=Molecular Biology Resources (8/05)

R=Promega Corporation (9/05)

S=Sigma Chemical Corporation (9/05)

U=Bangalore Genei (9/05)

V=Vivantis Technologies (1/06)

X=EURx Ltd. (9/05) Y=CinnaGen Inc. (9/05)

Parameters

Input	
Sequence	Name of the input FASTA file
Output	
Result File	Name of the output file
Commercial sites	Print additional table with commercial sites only
XML data	Name of the output file
Options	
Chain	Scan target sequence in different chain: **In direct chain only (default)** **In reverse chain only** **In both chains**
Recognition Site Length	Only enzymes with recognition sites equal to or greater than X bases long.
Restriction list	List of the restriction sites, use space as delimeter

SeqStat

Simple sequence statistics.
Parameters:

Input

Sequence	Name of the input file.
Output	
Result	Name of the output file.

SeqTrans

Simple sequence translate
Parameters:

Input	
Sequence	Name of the input file.
Output	
Result	Name of the output file.
Options	
ORF type	ORF type: **Full translation** - *translation of complete nucleotide sequences. As a result of performance of a command ("show output") translation in all given frameworks and chains will be received. **Longest frame** - *to give out the longest aminoacid sequence which is ends by stop-codon**. As a result of performance of a command the found sequence and full translation in a framework (and chain) for which sequence is found will be received. **Longest frame start with ATG** - * to give out the longest aminoacid sequence which begins with ATG ** and it is ends by stop-codon**. As a result of performance of a command the found sequence and full translation in a framework (and chain) for which sequence is found will be received.
Translation table	Translation table: Standart (1) Vertebrate Mitochondrial (2) Yeast Mitochondrial (3) Protozoan Mitochondrial and other (4) Invertebrate Mitochondrial (5) Ciliate Nuclear and other (6) Echinodermata Nuclear (9) Euplotid Nuclear (10) Bacterial (11) Alternative Yeast Nuclear (12) Ascidian Mitochondrial (13) Flatworm Mitochondrial (14) Blepharisma Macronuclear (15)

*Translation and search after translation is conducted only in the given chains and frameworks. For example, if the direction of a chain (+/-) and translation in the first framework is chosen, translation and search after translation will be made only for the first framework in (+) and (-) chains.

** in nucleotide sequence.

Statistics

F-test.

The program performs *F*-test for significantly different variances. The test trying to reject the null hypothesis that variances of two distributions are actually consistent. The statistic *F* is the ratio of one variance to the other. The values of the statistic either >> 1 or <<1 will indicate very significant differences. The null hypothesis (of equal variances) is trying to be rejected by either very large or very small values of *F*, so the significance is two-tailed.

This program use statistical functions from "R" free software environment for statistical computing and graphics (http://www.r-project.org).

This program requires the R-package to be installed on your computer.

Input file should contain table of numerical data: lines for observations (cases) columns should be separated by tabulation or user-defines symbol (; , etc); for example, if comma (,) separator is used, the file format is the same as the CSV (comma separated values) format. No missing data allowed.

Example of output data:

```
F-test for varince difference (two-tailed):
VarName M       Var
Feat1   -2.6040 101.8692
Feat5   2.0072  102.6015
F-statistics    1.0072
df1     49
df2     49
prob    0.9801
```

First line is the header. Second line prints data descriptions, separated by tabulation (VarName - names for selected variables; M - mean values for variables; Var - variances for variables). Next lines are the list data for variables (names, means and variances), separated by tabulation. After the variable list the following parameters are printed out: Pooled Variance (PooledVariance), F-statistics, number of degrees of freedom for variables (df1 and df2) and the probability the value of *F*-statistics under the null hypothesis of equal variances (prob).

Example of input data file format:

ItemName	Feat1	Feat2	Feat3	Feat4	Feat5	ClassVar
Item1	-11.761101	-5.295846	-2.491684	4.151158	9.777093	1
Item2	-11.425886	-6.753716	0.136692	5.161748	13.618702	1
Item3	-7.069796	0.545457	0.097140	0.678579	10.302988	1
Item4	-13.480880	-3.867702	0.119297	2.333842	10.992096	1
Item5	-9.707938	-2.597949	-2.329997	2.928526	8.441053	1
Item6	-10.013794	-2.165258	-3.169195	2.625904	10.611103	1
Item7	-9.057161	-4.766594	1.691733	1.655782	7.046236	1
Item8	-8.562761	-1.272652	-3.990204	2.286294	12.768212	1
Item9	-12.724631	-4.710623	-2.114719	2.812189	6.434645	1
Item10	-9.593738	-5.478652	-1.799524	4.306497	9.514756	1
Item11	-7.699759	-1.546648	-0.423322	4.889767	9.228675	1
Item12	-13.158116	-2.891354	0.595935	2.264199	12.004761	1
Item13	-10.509598	-3.414075	-1.962310	1.263863	10.199896	1
Item14	-6.547624	-3.594928	-2.117222	5.168950	10.838221	1
Item15	-12.375988	-3.130436	-2.169164	1.537614	11.112888	1
Item16	-12.953032	-2.805048	0.085116	3.303354	7.405194	1

```
Item17 -11.370708   -2.848384    -0.848201    3.885525     10.569231    1
Item18 -13.117222   -7.025575    1.406507     7.069338     12.230415    1
Item19 -11.573168   0.288003     -2.826167    4.397137     10.851711    1
Item20 -7.993835    -1.204352    -1.924345    0.829829     10.314768    1
Item21 -9.225135    -2.512925    -1.608051    1.420301     9.766411     1
Item22 -8.402783    -0.890500    3.189703     3.754479     7.481063     1
Item23 -9.888180    -3.345775    1.965667     2.906369     11.488815    1
Item24 -11.686270   -5.389477    2.556932     1.661153     9.717826     1
Item25 -12.599567   -0.266091    -3.936308    0.751762     10.405225    1
Item26 -11.365093   -1.919706    -0.458052    1.861843     9.521104     1
Item27 -11.027619   -2.944884    -2.792962    4.144322     7.958556     1
Item28 -11.795160   -6.769646    0.908383     1.005066     11.240333    1
Item29 -13.629933   0.674184     -3.386853    -0.095859    10.490432    1
Item30 -7.823298    -5.452589    -2.336894    1.919889     9.421125     1
Item31 6.360118     6.794549     4.168188     -4.492538    -12.297555   0
Item32 8.774682     0.492721     1.587909     -5.486587    -12.361278   0
Item33 7.768181     3.989776     3.289377     -0.895444    -13.067171   0
Item34 8.581133     2.922361     3.952544     -4.450362    -6.787133    0
Item35 6.176519     4.526292     -2.771599    -3.477187    -7.316202    0
Item36 11.539781    -0.892880    2.868221     -1.456557    -11.008881   0
Item37 11.743034    3.527726     -0.635792    -2.067965    -7.151524    0
Item38 10.527299    1.460768     0.862300     -1.967742    -8.819727    0
Item39 8.148808     5.157964     -0.916135    -1.551958    -9.467513    0
Item40 9.241432     1.483108     -0.981933    1.046571     -8.504166    0
Item41 9.444197     4.963927     1.127201     -0.523484    -9.102817    0
Item42 11.545396    4.604968     4.818171     -5.046815    -13.494675   0
Item43 11.890988    1.220710     -2.069796    -2.942747    -8.996673    0
Item44 11.810480    2.031465     2.987976     -5.699606    -10.026246   0
Item45 10.806543    5.275155     4.969420     -2.792596    -11.345561   0
Item46 10.261177    3.586077     3.340220     -3.339244    -7.795038    0
Item47 8.407544     2.887997     3.104312     -3.734519    -9.758477    0
Item48 7.317484     5.553850     1.618000     -2.525315    -13.613147   0
Item49 10.654500    2.579577     1.922452     -3.765160    -10.414136   0
Item50 6.940641     3.525834     -0.660756    -4.105869    -10.064455   0
```

Parameters:

	Input
Data	File with the data in TABLE format. File should contain table data: lines for observations (cases) and columns for variables; columns should be separated by tabulation or user-defines symbol (; , etc); no missed data allowed.
List of variables 1	Index of 1st variable to compare variances.
List of variables 2	Index of 2nd variable to compare variances.
	Output
Result	Name of output file
	Options
Field separation	Symbol or regular expression for separation variables in line; by default is ";".
Commentary line symbol	Commentary line symbol (if line starts from Commentary Symbol, then this line is ignored) ; by default - no commentary line
Flip file before	Flip file before processing

processing	
Take Observation names from 1st column in table	Take Observation names from 1st column in table or Generate Observation names (Observation1,Observation2).

K-Means

K-Means (K-means clustering). The data given from input file is clustered by the k-means method, which aims to partition the points into k groups such that the sum of squares from points to the assigned cluster centres is minimized. At the minimum, all cluster centres are at the mean of their Voronoi sets (the set of data points which are nearest to the cluster centre).

This program use statistical functions from "R" free software environment for statistical computing and graphics (http://www.r-project.org).

This program requires the R-package to be installed on your computer.

Input file should contain table of numerical data: lines for observations (cases) columns should be separated by tabulation or user-defines symbol (; , etc); for example, if comma (,) separator is used, the file format is the same as the CSV (comma separated values) format. No missing data allowed.

Example of input data file format:

ItemName	Feat1	Feat2	Feat3	Feat4	Feat5	ClassVar
Item1	-11.761101	-5.295846	-2.491684	4.151158	9.777093	1
Item2	-11.425886	-6.753716	0.136692	5.161748	13.618702	1
Item3	-7.069796	0.545457	0.097140	0.678579	10.302988	1
Item4	-13.480880	-3.867702	0.119297	2.333842	10.992096	1
Item5	-9.707938	-2.597949	-2.329997	2.928526	8.441053	1
Item6	-10.013794	-2.165258	-3.169195	2.625904	10.611103	1
Item7	-9.057161	-4.766594	1.691733	1.655782	7.046236	1
Item8	-8.562761	-1.272652	-3.990204	2.286294	12.768212	1
Item9	-12.724631	-4.710623	-2.114719	2.812189	6.434645	1
Item10	-9.593738	-5.478652	-1.799524	4.306497	9.514756	1
Item11	-7.699759	-1.546648	-0.423322	4.889767	9.228675	1
Item12	-13.158116	-2.891354	0.595935	2.264199	12.004761	1
Item13	-10.509598	-3.414075	-1.962310	1.263863	10.199896	1
Item14	-6.547624	-3.594928	-2.117222	5.168950	10.838221	1
Item15	-12.375988	-3.130436	-2.169164	1.537614	11.112888	1
Item16	-12.953032	-2.805048	0.085116	3.303354	7.405194	1
Item17	-11.370708	-2.848384	-0.848201	3.885525	10.569231	1
Item18	-13.117222	-7.025575	1.406507	7.069338	12.230415	1
Item19	-11.573168	0.288003	-2.826167	4.397137	10.851711	1
Item20	-7.993835	-1.204352	-1.924345	0.829829	10.314768	1
Item21	-9.225135	-2.512925	-1.608051	1.420301	9.766411	1
Item22	-8.402783	-0.890500	3.189703	3.754479	7.481063	1
Item23	-9.888180	-3.345775	1.965667	2.906369	11.488815	1
Item24	-11.686270	-5.389477	2.556932	1.661153	9.717826	1
Item25	-12.599567	-0.266091	-3.936308	0.751762	10.405225	1
Item26	-11.365093	-1.919706	-0.458052	1.861843	9.521104	1
Item27	-11.027619	-2.944884	-2.792962	4.144322	7.958556	1
Item28	-11.795160	-6.769646	0.908383	1.005066	11.240333	1
Item29	-13.629933	0.674184	-3.386853	-0.095859	10.490432	1
Item30	-7.823298	-5.452589	-2.336894	1.919889	9.421125	1

Item31	6.360118	6.794549	4.168188	-4.492538	-12.297555	0
Item32	8.774682	0.492721	1.587909	-5.486587	-12.361278	0
Item33	7.768181	3.989776	3.289377	-0.895444	-13.067171	0
Item34	8.581133	2.922361	3.952544	-4.450362	-6.787133	0
Item35	6.176519	4.526292	-2.771599	-3.477187	-7.316202	0
Item36	11.539781	-0.892880	2.868221	-1.456557	-11.008881	0
Item37	11.743034	3.527726	-0.635792	-2.067965	-7.151524	0
Item38	10.527299	1.460768	0.862300	-1.967742	-8.819727	0
Item39	8.148808	5.157964	-0.916135	-1.551958	-9.467513	0
Item40	9.241432	1.483108	-0.981933	1.046571	-8.504166	0
Item41	9.444197	4.963927	1.127201	-0.523484	-9.102817	0
Item42	11.545396	4.604968	4.818171	-5.046815	-13.494675	0
Item43	11.890988	1.220710	-2.069796	-2.942747	-8.996673	0
Item44	11.810480	2.031465	2.987976	-5.699606	-10.026246	0
Item45	10.806543	5.275155	4.969420	-2.792596	-11.345561	0
Item46	10.261177	3.586077	3.340220	-3.339244	-7.795038	0
Item47	8.407544	2.887997	3.104312	-3.734519	-9.758477	0
Item48	7.317484	5.553850	1.618000	-2.525315	-13.613147	0
Item49	10.654500	2.579577	1.922452	-3.765160	-10.414136	0
Item50	6.940641	3.525834	-0.660756	-4.105869	-10.064455	0

Parameters:

Input	
Data	File with the data in TABLE format. File should contain table data: lines for observations (cases) and columns for variables; columns should be separated by tabulation or user-defines symbol (; , etc); no missed data allowed.
List of variables	List of variables for which calculate variances, namely column indices. If ALL specified, program use all variables for analysis. Examples of input: 1;2;3-7;12; 1-12; ALL If 'Observation name' parameter is set on, variable list should not contain 1.
Output	
Result	Name of output file
Options	
Field separation	Symbol or regular expression for separation variables in line; by default is ";".
Commentary line symbol	Commentary line symbol (if line starts from Commentary Symbol, then this line is ignored) ; by default - no commentary line
Number of cluster	Number of clusters or a set of initial (distinct).
Flip file before processing	Flip file before processing
Take Observation names from 1st column in table	Take Observation names from 1st column in table or Generate Observation names (Observation1,Observation2)

LDAClass

The program performs linear discriminant classification. The Linear Discriminant is commonly used techniques for data classification. For each data item the program calculates the value of the Linear Discriminant Function (LDF) obtained by LDAClass procedure and separate data into two groups depending on whether the value of LDF is greater or less than 0. The set of variables used for the LDF calculation should coincide with the set used to obtain LDF by LDAStat procedure.

This program use statistical functions from "R" free software environment for statistical computing and graphics (http://www.r-project.org).

This program requires the R-package to be installed on your computer.

File should contain table of numerical data: lines for observations (cases) columns should be separated by tabulation or user-defines symbol (; , etc); for example, if comma (,) separator is used, the file format is the same as the CSV (comma separated values) format. No missing data allowed.

Example of output data:

```
LDA Classification:
Case# CaseName    LDF   Class
1     Case 1      119.0071    1
2     Case 2      144.7172    1
3     Case 3      93.3094     1
4     Case 4      134.6366    1
5     Case 5      -118.9141   0
6     Case 6      -89.0323    0
7     Case 7      -87.1935    0
8     Case 8      -123.9162   0
```

First line is the header. Second line is the data description, separated by tabulation (Case # - case number, CaseName – case name, LDF – the value of the linear discriminant function for the case, Class – classification index. Next lines provide parameters for each case.

Example of input data file format:

ItemName	Feat1	Feat2	Feat3	Feat4	Feat5	ClassVar
Item1	-11.761101	-5.295846	-2.491684	4.151158	9.777093	1
Item2	-11.425886	-6.753716	0.136692	5.161748	13.618702	1
Item3	-7.069796	0.545457	0.097140	0.678579	10.302988	1
Item4	-13.480880	-3.867702	0.119297	2.333842	10.992096	1
Item5	-9.707938	-2.597949	-2.329997	2.928526	8.441053	1
Item6	-10.013794	-2.165258	-3.169195	2.625904	10.611103	1
Item7	-9.057161	-4.766594	1.691733	1.655782	7.046236	1
Item8	-8.562761	-1.272652	-3.990204	2.286294	12.768212	1
Item9	-12.724631	-4.710623	-2.114719	2.812189	6.434645	1
Item10	-9.593738	-5.478652	-1.799524	4.306497	9.514756	1
Item11	-7.699759	-1.546648	-0.423322	4.889767	9.228675	1
Item12	-13.158116	-2.891354	0.595935	2.264199	12.004761	1
Item13	-10.509598	-3.414075	-1.962310	1.263863	10.199896	1
Item14	-6.547624	-3.594928	-2.117222	5.168950	10.838221	1
Item15	-12.375988	-3.130436	-2.169164	1.537614	11.112888	1
Item16	-12.953032	-2.805048	0.085116	3.303354	7.405194	1
Item17	-11.370708	-2.848384	-0.848201	3.885525	10.569231	1
Item18	-13.117222	-7.025575	1.406507	7.069338	12.230415	1
Item19	-11.573168	0.288003	-2.826167	4.397137	10.851711	1
Item20	-7.993835	-1.204352	-1.924345	0.829829	10.314768	1
Item21	-9.225135	-2.512925	-1.608051	1.420301	9.766411	1
Item22	-8.402783	-0.890500	3.189703	3.754479	7.481063	1
Item23	-9.888180	-3.345775	1.965667	2.906369	11.488815	1
Item24	-11.686270	-5.389477	2.556932	1.661153	9.717826	1

Item25	-12.599567	-0.266091	-3.936308	0.751762	10.405225	1
Item26	-11.365093	-1.919706	-0.458052	1.861843	9.521104	1
Item27	-11.027619	-2.944884	-2.792962	4.144322	7.958556	1
Item28	-11.795160	-6.769646	0.908383	1.005066	11.240333	1
Item29	-13.629933	0.674184	-3.386853	-0.095859	10.490432	1
Item30	-7.823298	-5.452589	-2.336894	1.919889	9.421125	1
Item31	6.360118	6.794549	4.168188	-4.492538	-12.297555	0
Item32	8.774682	0.492721	1.587909	-5.486587	-12.361278	0
Item33	7.768181	3.989776	3.289377	-0.895444	-13.067171	0
Item34	8.581133	2.922361	3.952544	-4.450362	-6.787133	0
Item35	6.176519	4.526292	-2.771599	-3.477187	-7.316202	0
Item36	11.539781	-0.892880	2.868221	-1.456557	-11.008881	0
Item37	11.743034	3.527726	-0.635792	-2.067965	-7.151524	0
Item38	10.527299	1.460768	0.862300	-1.967742	-8.819727	0
Item39	8.148808	5.157964	-0.916135	-1.551958	-9.467513	0
Item40	9.241432	1.483108	-0.981933	1.046571	-8.504166	0
Item41	9.444197	4.963927	1.127201	-0.523484	-9.102817	0
Item42	11.545396	4.604968	4.818171	-5.046815	-13.494675	0
Item43	11.890988	1.220710	-2.069796	-2.942747	-8.996673	0
Item44	11.810480	2.031465	2.987976	-5.699606	-10.026246	0
Item45	10.806543	5.275155	4.969420	-2.792596	-11.345561	0
Item46	10.261177	3.586077	3.340220	-3.339244	-7.795038	0
Item47	8.407544	2.887997	3.104312	-3.734519	-9.758477	0
Item48	7.317484	5.553850	1.618000	-2.525315	-13.613147	0
Item49	10.654500	2.579577	1.922452	-3.765160	-10.414136	0
Item50	6.940641	3.525834	-0.660756	-4.105869	-10.064455	0

Parameters:

Input	
Data	File with the data in TABLE format. File should contain table data: lines for observations (cases) and columns for variables; columns should be separated by tabulation or user-defines symbol (; , etc); no missed data allowed.
Classification rules	Name of input file with classification rules
Output	
Result	Name of output file
Options	
Field separation	Symbol or regular expression for separation variables in line; by default is ";".
Commentary line symbol	Commentary line symbol (if line starts from Commentary Symbol, then this line is ignored) ; by default - no commentary line
Take Observation names from 1st column in table	Take Observation names from 1st column in table or Generate Observation names (Observation1,Observation2)
Flip file before processing	Flip file before processing

LDAStat

The program calculates Linear Discriminant Analysis (LDA) parameters using the train data separated onto two classes. The Linear Discriminant Analysis is commonly used techniques for data classification. This method maximizes the ratio of between-class variance to the within-class variance in dataset thereby guaranteeing maximal separability. The approach calculates Linear Discriminant Function (LDF) which coefficients are chosen so that they result in the best separation among the groups for train data set. Variables for the classification should be specified by the user; classes for the data should be specified in the ClassVar variable by 0 or 1 values.

The LDF can be applied in the LDAClass procedure to separate any data into two groups depending on whether the value of LDF is greater or less than 0.

This program use statistical functions from "R" free software environment for statistical computing and graphics (http://www.r-project.org).

This program requires the R-package to be installed on your computer.

File should contain table of numerical data: lines for observations (cases) columns should be separated by tabulation or user-defines symbol (; , etc); for example, if comma (,) separator is used, the file format is the same as the CSV (comma separated values) format. No missing data allowed.

Example of output data:

```
LDA Statistics for class variable ClassVar:
NCASES=50; NCLASS0=20; NCLASS1=30
Var   Mean0 Mean1 LDF
Feat1 9.3970      -10.6047    -5.0675
Feat2 3.2846      -3.1118     -0.6547
Feat3 1.6290      -0.9977     1.0895
Feat4 -2.9638     2.7626      1.1494
Feat5 -10.0696    10.0585     5.8385
B0    *     *     -3.1990
```

First line is the header. Second line is the sample description: NCASES – number of cases total; NCLASS0 – number of class 0 cases; NCLASS1 – number of class 1 cases. Next line is output data description: Var – name of variable; Mean0 – mean for class 0; mMean1 – mean for class 1; LDF – coefficient of the linear discriminant function for the variable and b0 coefficient (B0).

Example of input data file format:

ItemName	Feat1	Feat2	Feat3	Feat4	Feat5	ClassVar
Item1	-11.761101	-5.295846	-2.491684	4.151158	9.777093	1
Item2	-11.425886	-6.753716	0.136692	5.161748	13.618702	1
Item3	-7.069796	0.545457	0.097140	0.678579	10.302988	1
Item4	-13.480880	-3.867702	0.119297	2.333842	10.992096	1
Item5	-9.707938	-2.597949	-2.329997	2.928526	8.441053	1
Item6	-10.013794	-2.165258	-3.169195	2.625904	10.611103	1
Item7	-9.057161	-4.766594	1.691733	1.655782	7.046236	1
Item8	-8.562761	-1.272652	-3.990204	2.286294	12.768212	1
Item9	-12.724631	-4.710623	-2.114719	2.812189	6.434645	1
Item10	-9.593738	-5.478652	-1.799524	4.306497	9.514756	1
Item11	-7.699759	-1.546648	-0.423322	4.889767	9.228675	1
Item12	-13.158116	-2.891354	0.595935	2.264199	12.004761	1
Item13	-10.509598	-3.414075	-1.962310	1.263863	10.199896	1
Item14	-6.547624	-3.594928	-2.117222	5.168950	10.838221	1
Item15	-12.375988	-3.130436	-2.169164	1.537614	11.112888	1
Item16	-12.953032	-2.805048	0.085116	3.303354	7.405194	1
Item17	-11.370708	-2.848384	-0.848201	3.885525	10.569231	1
Item18	-13.117222	-7.025575	1.406507	7.069338	12.230415	1
Item19	-11.573168	0.288003	-2.826167	4.397137	10.851711	1
Item20	-7.993835	-1.204352	-1.924345	0.829829	10.314768	1
Item21	-9.225135	-2.512925	-1.608051	1.420301	9.766411	1
Item22	-8.402783	-0.890500	3.189703	3.754479	7.481063	1
Item23	-9.888180	-3.345775	1.965667	2.906369	11.488815	1

Item24	-11.686270	-5.389477	2.556932	1.661153	9.717826	1
Item25	-12.599567	-0.266091	-3.936308	0.751762	10.405225	1
Item26	-11.365093	-1.919706	-0.458052	1.861843	9.521104	1
Item27	-11.027619	-2.944884	-2.792962	4.144322	7.958556	1
Item28	-11.795160	-6.769646	0.908383	1.005066	11.240333	1
Item29	-13.629933	0.674184	-3.386853	-0.095859	10.490432	1
Item30	-7.823298	-5.452589	-2.336894	1.919889	9.421125	1
Item31	6.360118	6.794549	4.168188	-4.492538	-12.297555	0
Item32	8.774682	0.492721	1.587909	-5.486587	-12.361278	0
Item33	7.768181	3.989776	3.289377	-0.895444	-13.067171	0
Item34	8.581133	2.922361	3.952544	-4.450362	-6.787133	0
Item35	6.176519	4.526292	-2.771599	-3.477187	-7.316202	0
Item36	11.539781	-0.892880	2.868221	-1.456557	-11.008881	0
Item37	11.743034	3.527726	-0.635792	-2.067965	-7.151524	0
Item38	10.527299	1.460768	0.862300	-1.967742	-8.819727	0
Item39	8.148808	5.157964	-0.916135	-1.551958	-9.467513	0
Item40	9.241432	1.483108	-0.981933	1.046571	-8.504166	0
Item41	9.444197	4.963927	1.127201	-0.523484	-9.102817	0
Item42	11.545396	4.604968	4.818171	-5.046815	-13.494675	0
Item43	11.890988	1.220710	-2.069796	-2.942747	-8.996673	0
Item44	11.810480	2.031465	2.987976	-5.699606	-10.026246	0
Item45	10.806543	5.275155	4.969420	-2.792596	-11.345561	0
Item46	10.261177	3.586077	3.340220	-3.339244	-7.795038	0
Item47	8.407544	2.887997	3.104312	-3.734519	-9.758477	0
Item48	7.317484	5.553850	1.618000	-2.525315	-13.613147	0
Item49	10.654500	2.579577	1.922452	-3.765160	-10.414136	0
Item50	6.940641	3.525834	-0.660756	-4.105869	-10.064455	0

Parameters:

Input	
Data	File with the data in TABLE format. File should contain table data: lines for observations (cases) and columns for variables; columns should be separated by tabulation or user-defines symbol (; , etc); no missed data allowed.
List of variables	List of variables for which calculate variances, namely column indices. If ALL specified, program use all variables for analysis. Examples of input: 1;2;3-7;12; 1-12; ALL If 'Observation name' parameter is set on, variable list should not contain 1.
Output	
Result	Name of output file
LDA Statistics	Output LDA Statistics file
Options	
Field separation	Symbol or regular expression for separation variables in line; by default is ";".
Commentary line symbol	Commentary line symbol (if line starts from Commentary Symbol, then this line is ignored) ; by default - no commentary line
Classification variable	Classification variable, in the table data this column should contain parameter's values (numerical or text), but the number of possible values

	should not exceed 10.
Flip file before processing	Flip file before processing
Take Observation names from 1st column in table	Take Observation names from 1st column in table or Generate Observation names (Observation1,Observation2)

Means

The program calculates means of the values in columns of data in table format.

This program use statistical functions from "R" free software environment for statistical computing and graphics (http://www.r-project.org).

This program requires the R-package to be installed on your computer.

Input file should contain table of numerical data: lines for observations (cases) columns should be separated by tabulation or user-defines symbol (; , etc); for example, if comma (,) separator is used, the file format is the same as the CSV (comma separated values) format. No missing data allowed.

Example of output data:

```
Variable    Mean
Feat1 -2.6040
Feat2 -0.5532
Feat3 0.0530
Feat4 0.4721
Feat5 2.0072
```

First line provides data description, separated by tabulation (Variable – names for selected variables; Mean – mean values for variables). Next are the lines list means for variables.

Example of input data file format:

ItemName	Feat1	Feat2	Feat3	Feat4	Feat5	ClassVar
Item1	-11.761101	-5.295846	-2.491684	4.151158	9.777093	1
Item2	-11.425886	-6.753716	0.136692	5.161748	13.618702	1
Item3	-7.069796	0.545457	0.097140	0.678579	10.302988	1
Item4	-13.480880	-3.867702	0.119297	2.333842	10.992096	1
Item5	-9.707938	-2.597949	-2.329997	2.928526	8.441053	1
Item6	-10.013794	-2.165258	-3.169195	2.625904	10.611103	1
Item7	-9.057161	-4.766594	1.691733	1.655782	7.046236	1
Item8	-8.562761	-1.272652	-3.990204	2.286294	12.768212	1
Item9	-12.724631	-4.710623	-2.114719	2.812189	6.434645	1
Item10	-9.593738	-5.478652	-1.799524	4.306497	9.514756	1
Item11	-7.699759	-1.546648	-0.423322	4.889767	9.228675	1
Item12	-13.158116	-2.891354	0.595935	2.264199	12.004761	1
Item13	-10.509598	-3.414075	-1.962310	1.263863	10.199896	1
Item14	-6.547624	-3.594928	-2.117222	5.168950	10.838221	1
Item15	-12.375988	-3.130436	-2.169164	1.537614	11.112888	1
Item16	-12.953032	-2.805048	0.085116	3.303354	7.405194	1
Item17	-11.370708	-2.848384	-0.848201	3.885525	10.569231	1
Item18	-13.117222	-7.025575	1.406507	7.069338	12.230415	1
Item19	-11.573168	0.288003	-2.826167	4.397137	10.851711	1
Item20	-7.993835	-1.204352	-1.924345	0.829829	10.314768	1
Item21	-9.225135	-2.512925	-1.608051	1.420301	9.766411	1
Item22	-8.402783	-0.890500	3.189703	3.754479	7.481063	1
Item23	-9.888180	-3.345775	1.965667	2.906369	11.488815	1
Item24	-11.686270	-5.389477	2.556932	1.661153	9.717826	1

Item25	-12.599567	-0.266091	-3.936308	0.751762	10.405225	1
Item26	-11.365093	-1.919706	-0.458052	1.861843	9.521104	1
Item27	-11.027619	-2.944884	-2.792962	4.144322	7.958556	1
Item28	-11.795160	-6.769646	0.908383	1.005066	11.240333	1
Item29	-13.629933	0.674184	-3.386853	-0.095859	10.490432	1
Item30	-7.823298	-5.452589	-2.336894	1.919889	9.421125	1
Item31	6.360118	6.794549	4.168188	-4.492538	-12.297555	0
Item32	8.774682	0.492721	1.587909	-5.486587	-12.361278	0
Item33	7.768181	3.989776	3.289377	-0.895444	-13.067171	0
Item34	8.581133	2.922361	3.952544	-4.450362	-6.787133	0
Item35	6.176519	4.526292	-2.771599	-3.477187	-7.316202	0
Item36	11.539781	-0.892880	2.868221	-1.456557	-11.008881	0
Item37	11.743034	3.527726	-0.635792	-2.067965	-7.151524	0
Item38	10.527299	1.460768	0.862300	-1.967742	-8.819727	0
Item39	8.148808	5.157964	-0.916135	-1.551958	-9.467513	0
Item40	9.241432	1.483108	-0.981933	1.046571	-8.504166	0
Item41	9.444197	4.963927	1.127201	-0.523484	-9.102817	0
Item42	11.545396	4.604968	4.818171	-5.046815	-13.494675	0
Item43	11.890988	1.220710	-2.069796	-2.942747	-8.996673	0
Item44	11.810480	2.031465	2.987976	-5.699606	-10.026246	0
Item45	10.806543	5.275155	4.969420	-2.792596	-11.345561	0
Item46	10.261177	3.586077	3.340220	-3.339244	-7.795038	0
Item47	8.407544	2.887997	3.104312	-3.734519	-9.758477	0
Item48	7.317484	5.553850	1.618000	-2.525315	-13.613147	0
Item49	10.654500	2.579577	1.922452	-3.765160	-10.414136	0
Item50	6.940641	3.525834	-0.660756	-4.105869	-10.064455	0

Parameters:

Input	
Data	File with the data in TABLE format. File should contain table data: lines for observations and columns for variables; columns should be separated by tabulation or user-defines sybol (; , etc); no missing data allowed.
List of variables	List of variables for which calculate variances, namely column indices. If ALL specified, program use all variables for analysis. Examples of input: 1;2;3-7;12; 1-12; ALL If 'Observation name' parameter is set on, variable list should not contain 1.
Output	
Result	Name of output file
Significant digits	Specifies the minimum number of significant digits to be printed in values.
XML data	Name of the file for graphical output.
Title	User-specified title of the graph plot.
Author	User-specified name of the graph author.
Comment	User-specified graph additional commentary line.
X axis name	User-specified graph X axis name.
Y axis name	User-specified graph Y axis name.
Options	

Field separation	Symbol for separation variables in line; by default tabulation and space.
Commentary line symbol	Commentary line symbol (if line starts from CommentSymbol, then this line is ignored)
Flip file before processing	Flip file before processing
Take Observation names from 1st column in table	Take Observation names from 1st column in table or Generate Observation names (Observation1,Observation2)

PCA

PCA is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension.

This program use statistical functions from "R" free software environment for statistical computing and graphics (http://www.r-project.org).

This program requires the R-package to be installed on your computer.

Input file should contain table of numerical data: lines for observations (cases) columns should be separated by tabulation or user-defines symbol (; , etc); for example, if comma (,) separator is used, the file format is the same as the CSV (comma separated values) format. No missing data allowed.

Example of input data file format:

ItemName	Feat1	Feat2	Feat3	Feat4	Feat5	ClassVar
Item1	-11.761101	-5.295846	-2.491684	4.151158	9.777093	1
Item2	-11.425886	-6.753716	0.136692	5.161748	13.618702	1
Item3	-7.069796	0.545457	0.097140	0.678579	10.302988	1
Item4	-13.480880	-3.867702	0.119297	2.333842	10.992096	1
Item5	-9.707938	-2.597949	-2.329997	2.928526	8.441053	1
Item6	-10.013794	-2.165258	-3.169195	2.625904	10.611103	1
Item7	-9.057161	-4.766594	1.691733	1.655782	7.046236	1
Item8	-8.562761	-1.272652	-3.990204	2.286294	12.768212	1
Item9	-12.724631	-4.710623	-2.114719	2.812189	6.434645	1
Item10	-9.593738	-5.478652	-1.799524	4.306497	9.514756	1
Item11	-7.699759	-1.546648	-0.423322	4.889767	9.228675	1
Item12	-13.158116	-2.891354	0.595935	2.264199	12.004761	1
Item13	-10.509598	-3.414075	-1.962310	1.263863	10.199896	1
Item14	-6.547624	-3.594928	-2.117222	5.168950	10.838221	1
Item15	-12.375988	-3.130436	-2.169164	1.537614	11.112888	1
Item16	-12.953032	-2.805048	0.085116	3.303354	7.405194	1
Item17	-11.370708	-2.848384	-0.848201	3.885525	10.569231	1
Item18	-13.117222	-7.025575	1.406507	7.069338	12.230415	1
Item19	-11.573168	0.288003	-2.826167	4.397137	10.851711	1
Item20	-7.993835	-1.204352	-1.924345	0.829829	10.314768	1
Item21	-9.225135	-2.512925	-1.608051	1.420301	9.766411	1
Item22	-8.402783	-0.890500	3.189703	3.754479	7.481063	1
Item23	-9.888180	-3.345775	1.965667	2.906369	11.488815	1
Item24	-11.686270	-5.389477	2.556932	1.661153	9.717826	1
Item25	-12.599567	-0.266091	-3.936308	0.751762	10.405225	1
Item26	-11.365093	-1.919706	-0.458052	1.861843	9.521104	1
Item27	-11.027619	-2.944884	-2.792962	4.144322	7.958556	1
Item28	-11.795160	-6.769646	0.908383	1.005066	11.240333	1
Item29	-13.629933	0.674184	-3.386853	-0.095859	10.490432	1
Item30	-7.823298	-5.452589	-2.336894	1.919889	9.421125	1

Item31	6.360118	6.794549	4.168188	-4.492538	-12.297555	0
Item32	8.774682	0.492721	1.587909	-5.486587	-12.361278	0
Item33	7.768181	3.989776	3.289377	-0.895444	-13.067171	0
Item34	8.581133	2.922361	3.952544	-4.450362	-6.787133	0
Item35	6.176519	4.526292	-2.771599	-3.477187	-7.316202	0
Item36	11.539781	-0.892880	2.868221	-1.456557	-11.008881	0
Item37	11.743034	3.527726	-0.635792	-2.067965	-7.151524	0
Item38	10.527299	1.460768	0.862300	-1.967742	-8.819727	0
Item39	8.148808	5.157964	-0.916135	-1.551958	-9.467513	0
Item40	9.241432	1.483108	-0.981933	1.046571	-8.504166	0
Item41	9.444197	4.963927	1.127201	-0.523484	-9.102817	0
Item42	11.545396	4.604968	4.818171	-5.046815	-13.494675	0
Item43	11.890988	1.220710	-2.069796	-2.942747	-8.996673	0
Item44	11.810480	2.031465	2.987976	-5.699606	-10.026246	0
Item45	10.806543	5.275155	4.969420	-2.792596	-11.345561	0
Item46	10.261177	3.586077	3.340220	-3.339244	-7.795038	0
Item47	8.407544	2.887997	3.104312	-3.734519	-9.758477	0
Item48	7.317484	5.553850	1.618000	-2.525315	-13.613147	0
Item49	10.654500	2.579577	1.922452	-3.765160	-10.414136	0
Item50	6.940641	3.525834	-0.660756	-4.105869	-10.064455	0

Parameters:

Input	
Data	File with the data in TABLE format. File should contain table data: lines for observations (cases) and columns for variables; columns should be separated by tabulation or user-defines symbol (; , etc); no missed data allowed.
List of variables	List of variables for which calculate variances, namely column indices. If ALL specified, program use all variables for analysis. Examples of input: 1;2;3-7;12; 1-12; ALL If 'Observation name' parameter is set on, variable list should not contain 1.
Output	
Result	Name of output file
Options	
Field separation	Symbol or regular expression for separation variables in line; by default is ";".
Commentary line symbol	Commentary line symbol (if line starts from Commentary Symbol, then this line is ignored) ; by default - no commentary line
Flip file before processing	Flip file before processing
Take Observation names from 1st column in table	Take Observation names from 1st column in table or Generate Observation names (Observation1,Observation2)

Pearson

The program calculates correlation coefficients between the values in columns of data in table format.

This program use statistical functions from "R" free software environment for statistical computing and graphics (http://www.r-project.org).

This program requires the R-package to be installed on your computer.

Input file should contain table of numerical data: lines for observations (cases) columns should be separated by tabulation or user-defines sybol (;, etc); for example, if comma (,) separator is used, the file format is the same as the CSV (comma separated values) format. No missing data allowed.

Example of output data:

```
Set1\Set2   Feat2 Feat3 Feat4 Feat5
Feat1 0.82  0.53  -0.84 -0.96
Feat2 1.00  0.38  -0.79 -0.84
```

First line contains variable names from list 2 starting from the second column and separated by tabulation. First column correspond to the first set of variables. The values of the correlation coefficients between variables from the first (lines) and second (columns) lists are presented.

Example of input data file format:

ItemName	Feat1	Feat2	Feat3	Feat4	Feat5	ClassVar
Item1	-11.761101	-5.295846	-2.491684	4.151158	9.777093	1
Item2	-11.425886	-6.753716	0.136692	5.161748	13.618702	1
Item3	-7.069796	0.545457	0.097140	0.678579	10.302988	1
Item4	-13.480880	-3.867702	0.119297	2.333842	10.992096	1
Item5	-9.707938	-2.597949	-2.329997	2.928526	8.441053	1
Item6	-10.013794	-2.165258	-3.169195	2.625904	10.611103	1
Item7	-9.057161	-4.766594	1.691733	1.655782	7.046236	1
Item8	-8.562761	-1.272652	-3.990204	2.286294	12.768212	1
Item9	-12.724631	-4.710623	-2.114719	2.812189	6.434645	1
Item10	-9.593738	-5.478652	-1.799524	4.306497	9.514756	1
Item11	-7.699759	-1.546648	-0.423322	4.889767	9.228675	1
Item12	-13.158116	-2.891354	0.595935	2.264199	12.004761	1
Item13	-10.509598	-3.414075	-1.962310	1.263863	10.199896	1
Item14	-6.547624	-3.594928	-2.117222	5.168950	10.838221	1
Item15	-12.375988	-3.130436	-2.169164	1.537614	11.112888	1
Item16	-12.953032	-2.805048	0.085116	3.303354	7.405194	1
Item17	-11.370708	-2.848384	-0.848201	3.885525	10.569231	1
Item18	-13.117222	-7.025575	1.406507	7.069338	12.230415	1
Item19	-11.573168	0.288003	-2.826167	4.397137	10.851711	1
Item20	-7.993835	-1.204352	-1.924345	0.829829	10.314768	1
Item21	-9.225135	-2.512925	-1.608051	1.420301	9.766411	1
Item22	-8.402783	-0.890500	3.189703	3.754479	7.481063	1
Item23	-9.888180	-3.345775	1.965667	2.906369	11.488815	1
Item24	-11.686270	-5.389477	2.556932	1.661153	9.717826	1
Item25	-12.599567	-0.266091	-3.936308	0.751762	10.405225	1
Item26	-11.365093	-1.919706	-0.458052	1.861843	9.521104	1
Item27	-11.027619	-2.944884	-2.792962	4.144322	7.958556	1
Item28	-11.795160	-6.769646	0.908383	1.005066	11.240333	1
Item29	-13.629933	0.674184	-3.386853	-0.095859	10.490432	1
Item30	-7.823298	-5.452589	-2.336894	1.919889	9.421125	1
Item31	6.360118	6.794549	4.168188	-4.492538	-12.297555	0
Item32	8.774682	0.492721	1.587909	-5.486587	-12.361278	0
Item33	7.768181	3.989776	3.289377	-0.895444	-13.067171	0
Item34	8.581133	2.922361	3.952544	-4.450362	-6.787133	0
Item35	6.176519	4.526292	-2.771599	-3.477187	-7.316202	0
Item36	11.539781	-0.892880	2.868221	-1.456557	-11.008881	0

Item37	11.743034	3.527726	-0.635792	-2.067965	-7.151524	0
Item38	10.527299	1.460768	0.862300	-1.967742	-8.819727	0
Item39	8.148808	5.157964	-0.916135	-1.551958	-9.467513	0
Item40	9.241432	1.483108	-0.981933	1.046571	-8.504166	0
Item41	9.444197	4.963927	1.127201	-0.523484	-9.102817	0
Item42	11.545396	4.604968	4.818171	-5.046815	-13.494675	0
Item43	11.890988	1.220710	-2.069796	-2.942747	-8.996673	0
Item44	11.810480	2.031465	2.987976	-5.699606	-10.026246	0
Item45	10.806543	5.275155	4.969420	-2.792596	-11.345561	0
Item46	10.261177	3.586077	3.340220	-3.339244	-7.795038	0
Item47	8.407544	2.887997	3.104312	-3.734519	-9.758477	0
Item48	7.317484	5.553850	1.618000	-2.525315	-13.613147	0
Item49	10.654500	2.579577	1.922452	-3.765160	-10.414136	0
Item50	6.940641	3.525834	-0.660756	-4.105869	-10.064455	0

Parameters:

Input	
Data	File with the data in TABLE format. File should contain table data: lines for observations and columns for variables; columns should be separated by tabulation or user-defines sybol (; , etc); no missing data allowed.
List of variables 1	List of variables for which calculate variances, namely column indices. If ALL specified, program use all variables for analysis. Examples of input: 1;2;3-7;12; 1-12; ALL If 'Observation name' parameter is set on, variable list should not contain 1.
List of variables 2	List of variables for which calculate variances, namely column indices. If ALL specified, program use all variables for analysis. Examples of input: 1;2;3-7;12; 1-12; ALL If 'Observation name' parameter is set on, variable list should not contain 1.
Output	
Result	Name of output file
Options	
Field separation	Symbol for separation variables in line; by default tabulation and space.
Commentary line symbol	Commentary line symbol (if line starts from CommentSymbol, then this line is ignored)
Flip file before processing	Flip file before processing
Take Observation names from 1st column in table	Take Observation names from 1st column in table or Generate Observation names (Observation1,Observation2)

R-Script

R-Script - enable running of the user's script, written in R language.
This program requires the R-package to be installed on your computer.

Parameters:

Input	
R-script	File whith R script.
Output	
Result	Name of output file

SNNBP-Learn

The program implements the function of learning multi-layer perceptron neural network.

Algorithm description.

The package implements the neural network of the multi-layer perceptron (MLP) topology.

MLP topology description.

The feed-forward neural network model transforms input signals into outputs. The transformation occurs at the neural network units called neurons (Fig. 1). The neuron consists of the weighted summation module (denoted as Σ in the Fig. 1) and non-linear transformation module (denoted as *F* in the Fig. 1). Such neuron structure is called perceptron.

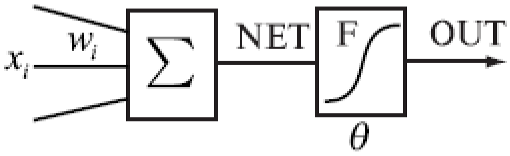


Fig. 1. The structure of the neuron.

NET is the result of the weighted summation of the input signals x_i. OUT is the output of the single neuron, and it is the result of the non-linear transformation by activation function *F* of the NET value.

$$NET=\sum_i w_i x_i$$
$$OUT=F(NET-\theta)$$

where

$\boldsymbol{x}=\{x_i\}$ – the input signals vector,

$\boldsymbol{w}=\{\mathrm{w}_i\}$ – weights,

θ - bias term,

F – neuron activation function,

NET-weighted sum of the input signals,

OUT – output signal.

The SNNBP program implements the feed-forward neural network where single units are connected in such way that output of one unit can be input to another unit. In the multi-layer perceptron topology units are combined in sets of layers with no connection of neurons within the layer. Neurons can input signals only from units of the previous layer and forward signals to the units of the next layer (Fig. 2). The number of neurons in the layer is arbitrary and set by user. The number of layers in the network is arbitrary (set by user).

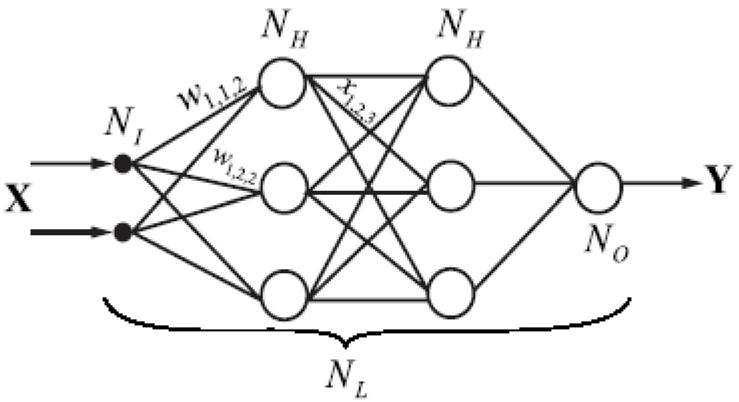


Fig. 2. The structure of the multi-layer perceptron.

There are three types of layers in such network. Fist is input layer, second is output layer, other layers called hidden. Neurons of the input layer make no transformations, they transmit the input signals to the first hidden layer. The SNNBP implements the algorithm that transformation of the the *i*-th neuron of the *k*-th layer as follows:

$$NET_{k,i} = \sum_{i=1}^{L_k} \sum_{j=1}^{L_{k-1}} w_{kij} OUT_{k-1,j} + w_{ki0} \quad ,$$
$$OUT_{k,i} = F\left(NET_{k,i}\right)$$

where $NET_{k,j}$ is the weighted sum of the inputs for the *i*-th neuron of the *k*-th layer (i=1,L_k, L_k – the number of neurons in the *k*-layer).

OUT$_{k,j}$ is the output value of the *i* –th neuron in the *k*-th layer.

$\boldsymbol{w}_{ki}$={w$_{kij}$} is the weight matrix, connecting the *i* –th neuron in the *k*-layer with the *j*–th neuron outputs (j=1,L_{k-1}) of the *k*-1 –th layer outputs.

w_{ki0} is the bias for the *i* 0th neuron in the *k*-th layer.

F is the activation function, the current version of the SNNBP program implement sigmoid activation function:

$$F = \frac{1}{1 + \exp\left(-NET \cdot c\right)} \quad ,$$

where *c* is the shape parameter (gain) that determines the slope of the sigmoid, when it is close to 0, the slope of the sigmoid is softer, if the gain is large, the shape is close to the step-wise function. The gain parameter is the same for all the neurons in the network.

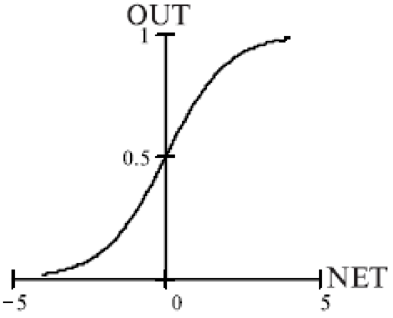


Fig. 3. The sigmoid activation function.

The SNNBP program allows setting the network topology of the arbitrary size of the input vector, output vector, number of hidden layers and number of neurons per layer. The network topology is set by user, as a rule, the topology can be optimized by trial and error procedure by user. The network with the simple structure may not capture the relationship between the input and output variables sufficiently. The multi-layer perceptron of the large size are more time-consuming to learn and need the large size of the training set to estimate the weights of the network. It is usual practice to start with the simple topology, then add more neurons and control the error after the topology changes.

The network model considers numerical representation of the input and output variables. It is able to solve the following types of tasks.
1). *The non-linear regression or prediction*. The neural network is trained to predict the output (target) values using the input value. In most cases, there is one (target) value at the neural network output tan need to be predicted. However multiple outputs can be predicted by SNNBP program also.
2). *Classification*. The neural network should classify the input sample by its input values into several classes. To code the classes several approaches exist. If it is needed to classify samples into 2 classes, the output of the network can be the single value and the classifying decision is determined by threshold value. Another way is to associate the class value to single output neuron and to select class according to the neuron with maximal output. The last method allows classifying samples into arbitrary number of classes.

The MLP learning procedure.
The idea behind the neural network is that the network can be trained to find the relationships between the input and output data. The learning process assumes the existence of the data for which the true relationship is known (supervised learning). The training data consist of samples for which the relationship between the inputs $\boldsymbol{x}$ and outputs $\boldsymbol{o}$ is known. For the specified network topology, learning procedure selects weights $\boldsymbol{w}_{ki}$ to minimize error between the outputs of the network and the true output values $\boldsymbol{t}$ (targets).
For the single sample n the targets $\boldsymbol{t}$ are known and the outputs $\boldsymbol{o}$ of the network are calculated (the size of the output and target vectors are equal to M), then the error can be estimated as follows:

$$E_n = \frac{1}{2}\sum_{m=1}^{M}(o_{nm} - t_{nm})^2 \ .$$

For the N samples total error estimate is

$$E = \sum_{n=1}^{N} E_n \ .$$

The learning task for the neural network is formulated as to fond the network topology and corresponding network parameters (weights) with the minimal value E for some training data set. This is the optimization problem. For neural network it can be solved numerically by steepest gradient method. The overall optimization scheme is as follows:
1). Set initial weight values if the MLP by random values [–0.5; 0.5].
2). Calculate the gradient direction.
3). Change the weight values w_{kij} (and biases w_{ki0}) for the $\alpha \cdot d_{kij}$, where α - is the step length (learning rate), d_{kij} is the vector of anti-gradient.
4). Repeat steps 2-3 until the error changes during optimization procedure will be small enough.

The SNNBP program implement slightly different optimization based on the error back-propagation algorithm. This is convenient and fast way for gradient calculation. This algorithm allow to calculate weight changes backward, from last layer to the first, the weights for the L_k level are calculated using the error estimates for the neurons in the L_{k+1} level. This allows to calculate all the weight changes recursively. The estimate of the gradient is possible in such a way that samples presented to the neural network sequentially. The learning process is divided to the "epochs", during the epoch all the samples from the training data are presented to the neural network. This is so-called batch training option.
The learning algorithm work as follows.
1). Set initial weight values if the MLP by random values [–0.5; 0.5].
2). Present the sample n from the training data to the network.
3). Calculate the outputs $\boldsymbol{o}$ of the NN for the inputs $\boldsymbol{x}$ of the sample.
4). Calculate the error between the outputs $\boldsymbol{o}$ and targets $\boldsymbol{t}$ for the sample n.

5). Using the backpropagation algorithm estimates the gradient are calculated and change the neural network weights according the gradient values are made.

6). Repeat steps 2-5 for all the samples from the training data.
In this procedure, samples are presented to the network randomly during the epoch. The overall learning cycle consisted of the several epochs usually. The number of epochs per learning step is defined by user and selected by trial and error procedure.

Momentum.
Usually, the gradient vector is estimated for current values of the network weights. The step length in the anti-gradient direction is α. In some cases the optimization efficiency can be improved by adding to the descent vector at the current step the vector at the previous step with some coefficient (momentum). This allows searching optimum efficiently in the narrow ravines of the error surfaces. In this case the weight w_{kij} changes (and w_{ki0}) made by the value $\alpha\cdot(d_{kij}$ + d_{kij}(previous)*m), where α - descent step length (learning rate), d_{kij} is the gradient direction at the current step, d_{kij}(previous) is the anti-gradient direction at the previous step, m is momentum (ranges from 0 to 1). If the moment is equal to 0, the descent direction vector is determined from the current weight values.

The learning protocol with early stopping.
If the network topology contains many weight parameters, it can over-fit the data in the learning process. This means that the network can recognize the data on which it was trained and cannot make generalizations for another data. This occur when the training data size is insufficient to fit the large number of parameters. To overcome the problem the early stopping procedure is implemented in the course of learning.
The protocol requires additional set of data, validating data set. These data serve as additional check for stop learning process, if the error became increasing on the validating data. The protocol for earsly stopping is as follows.
1). The number of training steps Nsteps is set.
2). At the each step the process of the learning by user-defined number of epochs is performed as described previously.
3). After each step the error of the NN is estimated on the validating data. If the error is less than was obtained previously, the network parameters are saved.
4). Otherwise the learning process continues until the number of learning steps is less than Nsteps or the error on the validating data is too large (say, 2 times larger than the minimal error obtained in previous steps). This process always saves the network parameters, which give the minimal error obtained during learning process for the validating data. The threshold parameter for large error deviation is set by the user.
The error on the training data in this protocol usually decreases to the small value and became fluctuating after some steps of learning. The error on the validating data is also decreasing after some steps, but at some point it may became increasing (the point where over-fitting occur). This protocol allows overcoming the over-fitting problem efficiently.

The SNNBP options.
The SNNBP program allows three options: learning, testing and prediction.
First option (*SNNBP –Learn*) implement the back-propagation training algorithm and output the optimal NN structure, saved in the SNNBP internal format. It is also possible to save the network parameters in the C file that can be compiled as a separate module that implements the NN evaluation by C-function. It also implement some additional features:

Internal normalization. After reading all the data are normalized in such a way that variables are scaled to the interval [0.1;0.9]. There is no need in data normalization by user. The neural network prediction values are rescaled back after prediction to the initial data range.

Prediction output. The program may save predicted values obtained by best network parameters for the training, validating and the testing data.

Second, testing option (*SNNBP-Test*) implement testing of the previously obtained network on the user data. The file should contain both input and output values. The error estimate is printed out. User can also output predicted values (outputs) for test data into user-defined file.

Third, prediction option (*SNNBP-Predict*) is implemented. In this option neural network calculate output values (predictions) using input values from the data file (target values need not be specified in this option). The predicted values are saved into user-defined file. The error is not calculated in this option.

Parameter description

	Input
Training data	File should contain table data: lines for observations (samples) and columns for variables; columns should be separated by tabulation or user-defines sybol (;, etc); no missed data allowed. The training data is mandatory parameter.
Testing data	File should contain table data: lines for observations (samples) and columns for variables; columns should be separated by tabulation or user-defines sybol (;, etc); no missed data allowed. The training data is not mandatory parameter, if it is omitted, the testing will be performed on the training data.
Validating data	File should contain table data: lines for observations (samples) and columns for variables; columns should be separated by tabulation or user-defines sybol (;, etc); no missed data allowed. The validating data is not mandatory parameter, if it is omitted, the validating will be performed on the training data.
Structure	Recently obtained file with network parameters to start from this network. To continue training network from previously saved parameters the network structure file in MLP format can be specified. This parameter is optional. If it is not stated, the learning begins with random NN weights.
List of input variables	List of variables which serve as predictors for NN, the input of the neural network. Examples of input: 1;2;3-7;12; 1-12; ALL If 'Observation name' parameter is set on, variable list should not contain 1.
List of target variables	List of targets variables (to be predicted by neural network). Format of input: 1;2;3-7;12; ALL
	Output
Status	Output file with the calculation status
Network structure	Output file with network structure and parameters in MLP format. This file can be used for prediction by neural network algorithm in snnbp.
Format in C-code file	Numerical format in C-code file. The format for weight data representation in C-code file. This is numerical (c-like, but without %) format for prediction output. Example: for .3 format the output will be presented as ...NNNN.NNN (where N - decimal numeral).
C-data	File to save neural network data as C function. The network parameters could be saved as C-code file. The parameter is optional. If it is not set, no C-code file will be generated.
Prediction	If this parameter is set ON, for each of the training/testing/validation file

output option	additional file with *.pred extention will be created containing predicted and observed values of the output variables.
Options	
Significant digits	String in C-type format description (without %), examples: 5.3f; .5f; 3.0f
Check names of variables	Check names of variables from table first row: **Take 1-st line in the table** **Take 1-st line in the table**
Check names of samples	Check names of samples from table first column: **Take 1-st line in the table** **Take 1-st line in the table**
Column separation	Symbol for separation variables in line; by default tabulation and space.
Commentary line symbol	Commentary line symbol (if line starts from CommentSymbol, then this line is ignored)
Number of layers	Number of layers in the neural network, including input and layers
Hidden layers sizes	Number of neurons in each hidden layer separated by semicolon. Example: 10;3; for 10 neurons in 1st hidden layer and 3 neurons in the 2nd hidden layer.
Momentum	The momentum value
Learning rate	Learning rate
Gain	Gain, the slope of the sigmoid function in the non-linear transformation of the NN
Number of epochs	The number of epochs per trainig step in the learning process
Number of training steps	The number of training steps in the learning process
Threshold for large error deviation	This parameter specify the error threshold for learning stopping criteria. It meaning depend on the StopCriteria setting.
Stopping criteria	This parameter defines the criteria to stop learning process. **Zero** - if the error is 0 (default); **NSteps** - if the the error did not decreased last LargeErrDev steps; **Barrier** - if the error increases after reaching its minimum (min_err) and the error is min_err*LargeErrDev.
Error estimation source	This parameter specify on which data to estimate error for stopping criteria. **Validating** - for testing data; **Training** - for training data.
Sampling protocol	This parameter specify the sampling protocol. **RandTime** - random sampling and on-line training, random generator initialized from the timer; RandInit - the same as previous, but the initialization is from the internally defined integer; **Sequentially** - samples are presented sequentially from the data, batch trainin is performed.

SNNBP-Predict

The program implements the prediction by multi-layer perceptron neural network.

Algorithm description.

The package implements the neural network of the multi-layer perceptron (MLP) topology.

MLP topology description.

The feed-forward neural network model transforms input signals into outputs. The transformation occurs at the neural network units called neurons (Fig. 1). The neuron consists of the weighted summation module (denoted as Σ in the Fig. 1) and non-linear transformation module (denoted as *F* in the Fig. 1). Such neuron structure is called perceptron.

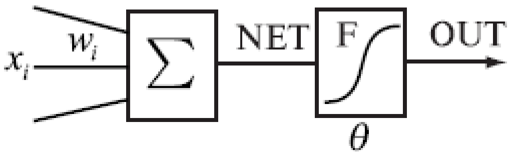


Fig. 1. The structure of the neuron.

NET is the result of the weighted summation of the input signals x_i. OUT is the output of the single neuron, and it is the result of the non-linear transformation by activation function *F* of the NET value.

$$NET=\sum_i w_i x_i$$
$$OUT=F(NET-\theta)$$

where
$\boldsymbol{x}=\{x_i\}$ – the input signals vector,
$\boldsymbol{w}=\{\mathrm{w}_i\}$ – weights,
θ - bias term,
F – neuron activation function,
NET-weighted sum of the input signals,
OUT – output signal.

The SNNBP program implements the feed-forward neural network where single units are connected in such way that output of one unit can be input to another unit. In the multi-layer perceptron topology units are combined in sets of layers with no connection of neurons within the layer. Neurons can input signals only from units of the previous layer and forward signals to the units of the next layer (Fig. 2). The number of neurons in the layer is arbitrary and set by user. The number of layers in the network is arbitrary (set by user).

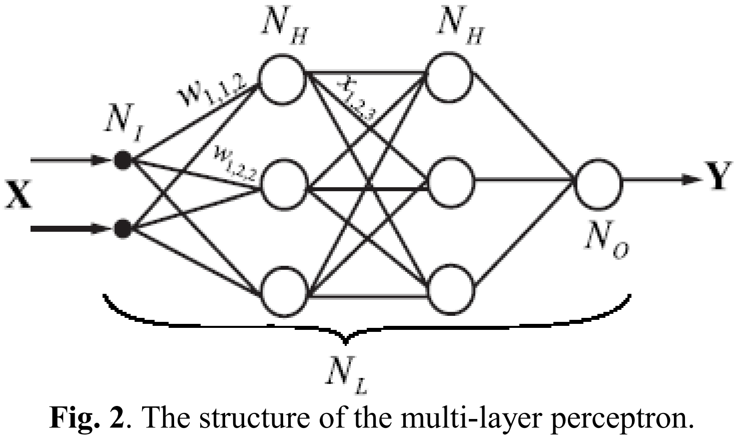


Fig. 2. The structure of the multi-layer perceptron.

There are three types of layers in such network. Fist is input layer, second is output layer, other layers called hidden. Neurons of the input layer make no transformations, they transmit the input signals to the first hidden layer. The SNNBP implements the algorithm that transformation of the the *i*-th neuron of the *k*-th layer as follows:

$$NET_{k,i}=\sum_{i=1}^{L_k}\sum_{j=1}^{L_{k-1}} w_{kij} OUT_{k-l,j}+w_{ki0} \quad ,$$
$$OUT_{k,i}=F(NET_{k,i})$$

where $NET_{k,j}$ is the weighted sum of the inputs for the *i*-th neuron of the *k*-th layer (i=1,L_k, L_k – the number of neurons in the *k*-layer).
OUT$_{k,j}$ is the output value of the *i* –th neuron in the *k*-th layer.
$\boldsymbol{w}_{ki}$={w$_{kij}$} is the weight matrix, connecting the *i* –th neuron in the *k*-layer with the *j*–th neuron outputs (j=1,L_{k-1}) of the *k*-1 –th layer outputs.
w_{ki0} is the bias for the *i* 0th neuron in the *k*-th layer.
F is the activation function, the current version of the SNNBP program implement sigmoid activation function:

$$F=\frac{1}{1+\exp(-NET\cdot c)},$$

where *c* is the shape parameter (gain) that determines the slope of the sigmoid, when it is close to 0, the slope of the sigmoid is softer, if the gain is large, the shape is close to the step-wise function. The gain parameter is the same for all the neurons in the network.

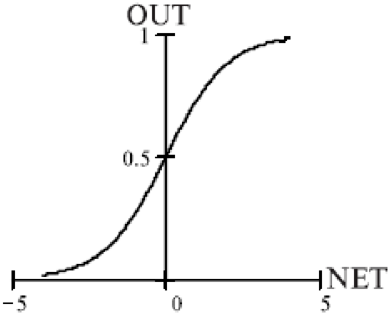


Fig. 3. The sigmoid activation function.

The SNNBP program allows setting the network topology of the arbitrary size of the input vector, output vector, number of hidden layers and number of neurons per layer. The network topology is set by user, as a rule, the topology can be optimized by trial and error procedure by user. The network with the simple structure may not capture the relationship between the input and output variables sufficiently. The multi-layer perceptron of the large size are more time-consuming to learn and need the large size of the training set to estimate the weights of the network. It is usual practice to start with the simple topology, then add more neurons and control the error after the topology changes.
The network model considers numerical representation of the input and output variables. It is able to solve the following types of tasks.
1). *The non-linear regression or prediction*. The neural network is trained to predict the output (target) values using the input value. In most cases, there is one (target) value at the neural network output tan need to be predicted. However multiple outputs can be predicted by SNNBP program also.
2). *Classification*. The neural network should classify the input sample by its input values into several classes. To code the classes several approaches exist. If it is needed to classify samples into 2 classes, the output of the network can be the single value and the classifying decision is determined by threshold value. Another way is to associate the class value to single output neuron and to select class according to the neuron with maximal output. The last method allows classifying samples into arbitrary number of classes.

The MLP learning procedure.
The idea behind the neural network is that the network can be trained to find the relationships between the input and output data. The learning process assumes the existence of the data for which the true relationship is known (supervised learning). The training data consist of samples for which the relationship between the inputs $\boldsymbol{x}$ and outputs $\boldsymbol{o}$ is known. For the specified network topology, learning procedure selects weights $\boldsymbol{w}_{ki}$ to minimize error between the outputs of the network and the true output values $\boldsymbol{t}$ (targets).

For the single sample *n* the targets ***t*** are known and the outputs ***o*** of the network are calculated (the size of the output and target vectors are equal to M), then the error can be estimated as follows:

$$E_n = \frac{1}{2} \sum_{m=1}^{M} (o_{nm} - t_{nm})^2 \ .$$

For the N samples total error estimate is

$$E = \sum_{n=1}^{N} E_n \ .$$

The learning task for the neural network is formulated as to fond the network topology and corresponding network parameters (weights) with the minimal value *E* for some training data set. This is the optimization problem. For neural network it can be solved numerically by steepest gradient method. The overall optimization scheme is as follows:

1). Set initial weight values if the MLP by random values [–0.5; 0.5].

2). Calculate the gradient direction.

3). Change the weight values w_{kij} (and biases w_{ki0}) for the $\alpha \cdot d_{kij}$, where α - is the step length (learning rate), d_{kij} is the vector of anti-gradient.

4). Repeat steps 2-3 until the error changes during optimization procedure will be small enough.

The SNNBP program implement slightly different optimization based on the error back-propagation algorithm. This is convenient and fast way for gradient calculation. This algorithm allow to calculate weight changes backward, from last layer to the first, the weights for the L_k level are calculated using the error estimates for the neurons in the L_{k+1} level. This allows to calculate all the weight changes recursively. The estimate of the gradient is possible in such a way that samples presented to the neural network sequentially. The learning process is divided to the “epochs”, during the epoch all the samples from the training data are presented to the neural network. This is so-called batch training option.

The learning algorithm work as follows.

1). Set initial weight values if the MLP by random values [–0.5; 0.5].

2). Present the sample *n* from the training data to the network.

3). Calculate the outputs ***o*** of the NN for the inputs ***x*** of the sample.

4). Calculate the error between the outputs ***o*** and targets ***t*** for the sample *n*.

5). Using the backpropagation algorithm estimates the gradient are calculated and change the neural network weights according the gradient values are made.

6). Repeat steps 2-5 for all the samples from the training data.

In this procedure, samples are presented to the network randomly during the epoch. The overall learning cycle consisted of the several epochs usually. The number of epochs per learning step is defined by user and selected by trial and error procedure.

Momentum.

Usually, the gradient vector is estimated for current values of the network weights. The step length in the anti-gradient direction is α. In some cases the optimization efficiency can be improved by adding to the descent vector at the current step the vector at the previous step with some coefficient (momentum). This allows searching optimum efficiently in the narrow ravines of the error surfaces. In this case the weight w_{kij} changes (and w_{ki0}) made by the value $\alpha \cdot (d_{kij} + d_{kij}(\text{previous}) * m)$, where α - descent step length (learning rate), d_{kij} is the gradient direction at the current step, d_{kij}(previous) is the anti-gradient direction at the previous step, *m* is momentum (ranges from 0 to 1). If the moment is equal to 0, the descent direction vector is determined from the current weight values.

The learning protocol with early stopping.

If the network topology contains many weight parameters, it can over-fit the data in the learning process. This means that the network can recognize the data on which it was trained and cannot make generalizations for another data. This occur when the training data size is insufficient to fit the large number of parameters. To overcome the problem the early stopping procedure is implemented in the course of learning.

The protocol requires additional set of data, validating data set. These data serve as additional check for stop learning process, if the error became increasing on the validating data. The protocol for earsly stopping is as follows.

1). The number of training steps Nsteps is set.

2). At the each step the process of the learning by user-defined number of epochs is performed as described previously.

3). After each step the error of the NN is estimated on the validating data. If the error is less than was obtained previously, the network parameters are saved.

4). Otherwise the learning process continues until the number of learning steps is less than Nsteps or the error on the validating data is too large (say, 2 times larger than the minimal error obtained in previous steps). This process always saves the network parameters, which give the minimal error obtained during learning process for the validating data. The threshold parameter for large error deviation is set by the user.

The error on the training data in this protocol usually decreases to the small value and became fluctuating after some steps of learning. The error on the validating data is also decreasing after some steps, but at some point it may became increasing (the point where over-fitting occur). This protocol allows overcoming the over-fitting problem efficiently.

The SNNBP options.

The SNNBP program allows three options: learning, testing and prediction.

First option (*SNNBP –Learn*) implement the back-propagation training algorithm and output the optimal NN structure, saved in the SNNBP internal format. It is also possible to save the network parameters in the C file that can be compiled as a separate module that implements the NN evaluation by C-function. It also implement some additional features:

> *Internal normalization.* After reading all the data are normalized in such a way that variables are scaled to the interval [0.1;0.9]. There is no need in data normalization by user. The neural network prediction values are rescaled back after prediction to the initial data range.
>
> *Prediction output.* The program may save predicted values obtained by best network parameters for the training, validating and the testing data.

Second, testing option (*SNNBP-Test*) implement testing of the previously obtained network on the user data. The file should contain both input and output values. The error estimate is printed out. User can also output predicted values (outputs) for test data into user-defined file.

Third, prediction option (*SNNBP-Predict*) is implemented. In this option neural network calculate output values (predictions) using input values from the data file (target values need not be specified in this option). The predicted values are saved into user-defined file. The error is not calculated in this option.

Parameter description

Input	
Testing data	File should contain table data: lines for observations (samples) and columns for variables; columns should be separated by tabulation or user-defined symbol (;, etc); no missed data allowed. The testing data is mandatory parameter, it should contain predicting (inputs), but may not contain output variables.
Structure	This is the name of previously obtained network parameter file in MLP format
List of input variables	List of variables which serve as predictors for NN, the input of the neural network. Format of input : 1;2;3-7;12;

Output	
Errors	Output file, will contain error estimates for the NN predictions
Predictions	File should contain table data: lines for observations (samples) and columns for variables; columns should be separated by tabulation or user-defines sybol (;, etc); no missed data allowed. The validating data is not mandatory parameter, if it is omitted, the validating will be performed on the training data.
Options	
Significant digits	String in C-type format description (without %), examples: 5.3f; .5f; 3.0f
Check names of variables	Check names of variables from table first row: **Take 1-st line in the table** **Take 1-st line in the table**
Check names of samples	Check names of samples from table first column: **Take 1-st line in the table** **Take 1-st line in the table**
Column separation	Symbol for separation variables in line; by default tabulation and space.
Commentary line 1st character	Commentary line symbol (if line starts from CommentSymbol, then this line is ignored); by default - no commentary line

SNNBP-Test

The program implements testing the prediction by multi-layer perceptron neural network.

Algorithm description.

The package implements the neural network of the multi-layer perceptron (MLP) topology.

MLP topology description.

The feed-forward neural network model transforms input signals into outputs. The transformation occurs at the neural network units called neurons (Fig. 1). The neuron consists of the weighted summation module (denoted as Σ in the Fig. 1) and non-linear transformation module (denoted as *F* in the Fig. 1). Such neuron structure is called perceptron.

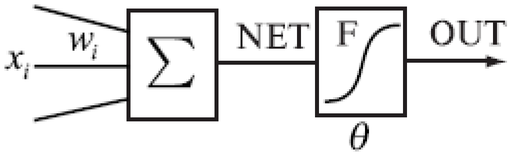


Fig. 1. The structure of the neuron.

NET is the result of the weighted summation of the input signals x_i. OUT is the output of the single neuron, and it is the result of the non-linear transformation by activation function *F* of the NET value.

$$NET = \sum_i w_i x_i$$
$$OUT = F(NET - \theta)$$

where

$\boldsymbol{x}=\{x_i\}$ – the input signals vector,

$\boldsymbol{w}=\{\mathrm{w}_i\}$ – weights,

θ - bias term,

F – neuron activation function,

NET-weighted sum of the input signals,

OUT – output signal.

The SNNBP program implements the feed-forward neural network where single units are connected in such way that output of one unit can be input to another unit. In the multi-layer perceptron topology units are combined in sets of layers with no connection of neurons within the layer. Neurons can input signals only from units of the previous layer and forward signals to the units of the next layer (Fig. 2). The number of neurons in the layer is arbitrary and set by user. The number of layers in the network is arbitrary (set by user).

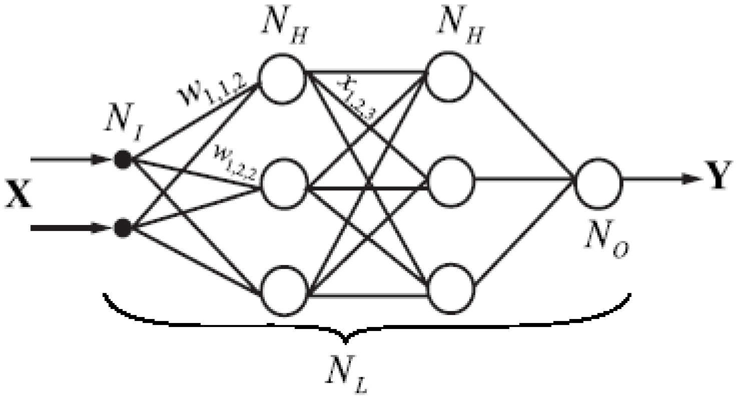


Fig. 2. The structure of the multi-layer perceptron.

There are three types of layers in such network. Fist is input layer, second is output layer, other layers called hidden. Neurons of the input layer make no transformations, they transmit the input signals to the first hidden layer. The SNNBP implements the algorithm that transformation of the the *i*-th neuron of the *k*-th layer as follows:

$$NET_{k,i} = \sum_{i=1}^{L_k} \sum_{j=1}^{L_{k-1}} w_{kij} OUT_{k-1,j} + w_{ki0} ,$$
$$OUT_{k,i} = F(NET_{k,i})$$

where $NET_{k,j}$ is the weighted sum of the inputs for the *i*-th neuron of the *k*-th layer (i=1,L_k, L_k – the number of neurons in the *k*-layer).

OUT$_{k,j}$ is the output value of the *i* –th neuron in the *k*-th layer.

$\mathbf{w}_{ki}$={w$_{kij}$} is the weight matrix, connecting the *i* –th neuron in the *k*-layer with the *j*–th neuron outputs (j=1,L_{k-1}) of the *k*-1 –th layer outputs.

w_{ki0} is the bias for the *i* 0th neuron in the *k*-th layer.

F is the activation function, the current version of the SNNBP program implement sigmoid activation function:

$$F = \frac{1}{1+\exp(-NET \cdot c)} ,$$

where *c* is the shape parameter (gain) that determines the slope of the sigmoid, when it is close to 0, the slope of the sigmoid is softer, if the gain is large, the shape is close to the step-wise function. The gain parameter is the same for all the neurons in the network.

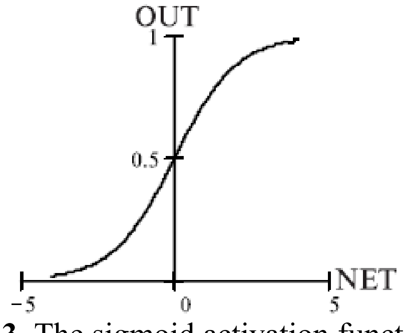


Fig. 3. The sigmoid activation function.

The SNNBP program allows setting the network topology of the arbitrary size of the input vector, output vector, number of hidden layers and number of neurons per layer. The network topology is set by user, as a rule, the topology can be optimized by trial and error procedure by user. The network with the simple structure may not capture the relationship between the input and output variables sufficiently. The multi-layer perceptron of the large size are more time-consuming to learn and need the large size of the training set to estimate the weights of the network. It is usual practice to start with the simple topology, then add more neurons and control the error after the topology changes.

The network model considers numerical representation of the input and output variables. It is able to solve the following types of tasks.

1). *The non-linear regression or prediction*. The neural network is trained to predict the output (target) values using the input value. In most cases, there is one (target) value at the neural network output tan need to be predicted. However multiple outputs can be predicted by SNNBP program also.

2). *Classification*. The neural network should classify the input sample by its input values into several classes. To code the classes several approaches exist. If it is needed to classify samples into 2 classes, the output of the network can be the single value and the classifying decision is determined by threshold value. Another way is to associate the class value to single output neuron and to select class according to the neuron with maximal output. The last method allows classifying samples into arbitrary number of classes.

The MLP learning procedure.

The idea behind the neural network is that the network can be trained to find the relationships between the input and output data. The learning process assumes the existence of the data for which the true relationship is known (supervised learning). The training data consist of samples for which the relationship between the inputs ***x*** and outputs ***o*** is known. For the specified network topology, learning procedure selects weights $\boldsymbol{w}_{ki}$ to minimize error between the outputs of the network and the true output values ***t*** (targets).

For the single sample *n* the targets ***t*** are known and the outputs ***o*** of the network are calculated (the size of the output and target vectors are equal to M), then the error can be estimated as follows:

$$E_n=\frac{1}{2}\sum_{m=1}^{M}(o_{nm}-t_{nm})^2 \ .$$

For the N samples total error estimate is

$$E=\sum_{n=1}^{N}E_n \ .$$

The learning task for the neural network is formulated as to fond the network topology and corresponding network parameters (weights) with the minimal value *E* for some training data set. This is the optimization problem. For neural network it can be solved numerically by steepest gradient method. The overall optimization scheme is as follows:

1). Set initial weight values if the MLP by random values [–0.5; 0.5].

2). Calculate the gradient direction.

3). Change the weight values w_{kij} (and biases w_{ki0}) for the $\alpha \cdot d_{kij}$, where α - is the step length (learning rate), d_{kij} is the vector of anti-gradient.

4). Repeat steps 2-3 until the error changes during optimization procedure will be small enough.

The SNNBP program implement slightly different optimization based on the error back-propagation algorithm. This is convenient and fast way for gradient calculation. This algorithm allow to calculate weight changes backward, from last layer to the first, the weights for the L_k level are calculated using the error estimates for the neurons in the $L_{\text{k}+1}$ level. This allows to calculate all the weight changes recursively. The estimate of the gradient is possible in such a way that samples presented to the neural network sequentially. The learning process is divided to the "epochs", during the epoch all the samples from the training data are presented to the neural network. This is so-called batch training option.

The learning algorithm work as follows.

1). Set initial weight values if the MLP by random values [–0.5; 0.5].

2). Present the sample *n* from the training data to the network.

3). Calculate the outputs ***o*** of the NN for the inputs ***x*** of the sample.

4). Calculate the error between the outputs ***o*** and targets ***t*** for the sample *n*.

5). Using the backpropagation algorithm estimates the gradient are calculated and change the neural network weights according the gradient values are made.

6). Repeat steps 2-5 for all the samples from the training data.

In this procedure, samples are presented to the network randomly during the epoch. The overall learning cycle consisted of the several epochs usually. The number of epochs per learning step is defined by user and selected by trial and error procedure.

Momentum.

Usually, the gradient vector is estimated for current values of the network weights. The step length in the anti-gradient direction is α. In some cases the optimization efficiency can be improved by adding to the descent vector at the current step the vector at the previous step with some coefficient (momentum). This allows searching optimum efficiently in the narrow ravines of the error surfaces. In this case the weight w_{kij} changes (and w_{ki0}) made by the value $\alpha\cdot(d_{kij} + d_{kij}(\text{previous})*m)$, where α - descent step length (learning rate), d_{kij} is the gradient direction at the current step, d_{kij}(previous) is the anti-gradient direction at the previous step, *m* is momentum (ranges from 0 to 1). If the moment is equal to 0, the descent direction vector is determined from the current weight values.

The learning protocol with early stopping.

If the network topology contains many weight parameters, it can over-fit the data in the learning process. This means that the network can recognize the data on which it was trained and cannot make generalizations for another data. This occur when the training data size is insufficient to fit the large number of parameters. To overcome the problem the early stopping procedure is implemented in the course of learning.

The protocol requires additional set of data, validating data set. These data serve as additional check for stop learning process, if the error became increasing on the validating data. The protocol for earsly stopping is as follows.

1). The number of training steps Nsteps is set.

2). At the each step the process of the learning by user-defined number of epochs is performed as described previously.

3). After each step the error of the NN is estimated on the validating data. If the error is less than was obtained previously, the network parameters are saved.

4). Otherwise the learning process continues until the number of learning steps is less than Nsteps or the error on the validating data is too large (say, 2 times larger than the minimal error obtained in previous steps). This process always saves the network parameters, which give the minimal error obtained during learning process for the validating data. The threshold parameter for large error deviation is set by the user.

The error on the training data in this protocol usually decreases to the small value and became fluctuating after some steps of learning. The error on the validating data is also decreasing after some steps, but at some point it may became increasing (the point where over-fitting occur). This protocol allows overcoming the over-fitting problem efficiently.

The SNNBP options.

The SNNBP program allows three options: learning, testing and prediction.

First option (*SNNBP –Learn*) implement the back-propagation training algorithm and output the optimal NN structure, saved in the SNNBP internal format. It is also possible to save the network parameters in the C file that can be compiled as a separate module that implements the NN evaluation by C-function. It also implement some additional features:

> *Internal normalization.* After reading all the data are normalized in such a way that variables are scaled to the interval [0.1;0.9]. There is no need in data normalization by user. The neural network prediction values are rescaled back after prediction to the initial data range.
>
> *Prediction output.* The program may save predicted values obtained by best network parameters for the training, validating and the testing data.

Second, testing option (*SNNBP-Test*) implement testing of the previously obtained network on the user data. The file should contain both input and output values. The error estimate is printed out. User can also output predicted values (outputs) for test data into user-defined file.

Third, prediction option (*SNNBP-Predict*) is implemented. In this option neural network calculate output values (predictions) using input values from the data file (target values need not be specified in this option). The predicted values are saved into user-defined file. The error is not calculated in this option.

Parameter description

Input	
Testing data	File should contain table data: lines for observations (samples) and columns for variables; columns should be separated by tabulation or user-defines sybol (;, etc); no missed data allowed. The testing data is mandatory parameter, it should contain both predicting (inputs) and predicted (outputs) variables.
Structure	This is the name of previously obtained network parameter file in MLP format
List of input variables	List of variables which serve as predictors for NN, the input of the neural network. Format of input : 1;2;3-7;12;
List of target variables	List of target variables (to be predicted by neural network). Format of input: 1;2;3-7;12; ALL
Output	
Errors	Output file, will contain error estimates for the NN predictions
Predictions	File should contain table data: lines for observations (samples) and columns for variables; columns should be separated by tabulation or user-defines sybol (;, etc); no missed data allowed. The validating data is not mandatory parameter, if it is omitted, the validating will be performed on the training data.
Options	
Significant digits	String in C-type format description (without %), examples: 5.3f; .5f; 3.0f
Check names of variables	Check names of variables from table first row: **Take 1-st line in the table** **Take 1-st line in the table**
Check names of samples	Check names of samples from table first column: **Take 1-st line in the table** **Take 1-st line in the table**
Column separation	Symbol for separation variables in line; by default tabulation and space.
Commentary line 1st character	Commentary line symbol (if line starts from CommentSymbol, then this line is ignored); by default - no commentary line

T-test.

The program performs Student’s *t*-test for significantly different means. This test is applied when two distributions *x* and *y* are thought to have the same variance, but possibly different means. The test evaluates the significance of the $t=(x_0-y_0)/\mathrm{SD}$, where x_0 and y_0 are mean estimates for *x* and *y*, SD is the “pooled variance”. The *t* value follows Student’s *t*-distribution with $N_x + N_y - 2$ degrees of freedom, where N_x and N_y are sample sizes for *x* and *y*. The significance is the probability that $|t|$ could be this large or larger just by chance, for distributions with equal means; a value of the significance smaller than, for example, 0.05 means that the observed difference is significant at 95% confidence.

This program use statistical functions from "R" free software environment for statistical computing and graphics (http://www.r-project.org).

This program requires the R-package to be installed on your computer.

Input file should contain table of numerical data: lines for observations (cases) columns should be separated by tabulation or user-defines symbol (; , etc); for example, if comma (,) separator is used, the file format is the same as the CSV (comma separated values) format. No missing data allowed.

Example of output data:

```
T-test for means difference (two-tailed):
VarName     M     Var
Feat1 -2.6040     101.8692
Feat5 2.0072      102.6015
PooledVariance    102.2353
t-statistics      2.2803
df    98
prob  0.0248
```

First line is the header. Second line is the data descriptions, separated by tabulation (VarName – names for selected variables; M – mean values for variables; Var – variances for variables). Next lines list data for variables (names, means and variances), separated by tabulation. After the variable list the following parameters are printed out: Pooled Variance (PooledVariance), *t*-statistics, number of degrees of freedom (df) and the probability that $|t|$ could be this large or larger just by chance (prob).

Example of input data file format:

ItemName	Feat1	Feat2	Feat3	Feat4	Feat5	ClassVar
Item1	-11.761101	-5.295846	-2.491684	4.151158	9.777093	1
Item2	-11.425886	-6.753716	0.136692	5.161748	13.618702	1
Item3	-7.069796	0.545457	0.097140	0.678579	10.302988	1
Item4	-13.480880	-3.867702	0.119297	2.333842	10.992096	1
Item5	-9.707938	-2.597949	-2.329997	2.928526	8.441053	1
Item6	-10.013794	-2.165258	-3.169195	2.625904	10.611103	1
Item7	-9.057161	-4.766594	1.691733	1.655782	7.046236	1
Item8	-8.562761	-1.272652	-3.990204	2.286294	12.768212	1
Item9	-12.724631	-4.710623	-2.114719	2.812189	6.434645	1
Item10	-9.593738	-5.478652	-1.799524	4.306497	9.514756	1
Item11	-7.699759	-1.546648	-0.423322	4.889767	9.228675	1
Item12	-13.158116	-2.891354	0.595935	2.264199	12.004761	1
Item13	-10.509598	-3.414075	-1.962310	1.263863	10.199896	1
Item14	-6.547624	-3.594928	-2.117222	5.168950	10.838221	1
Item15	-12.375988	-3.130436	-2.169164	1.537614	11.112888	1
Item16	-12.953032	-2.805048	0.085116	3.303354	7.405194	1
Item17	-11.370708	-2.848384	-0.848201	3.885525	10.569231	1
Item18	-13.117222	-7.025575	1.406507	7.069338	12.230415	1
Item19	-11.573168	0.288003	-2.826167	4.397137	10.851711	1

Item20	-7.993835	-1.204352	-1.924345	0.829829	10.314768	1
Item21	-9.225135	-2.512925	-1.608051	1.420301	9.766411	1
Item22	-8.402783	-0.890500	3.189703	3.754479	7.481063	1
Item23	-9.888180	-3.345775	1.965667	2.906369	11.488815	1
Item24	-11.686270	-5.389477	2.556932	1.661153	9.717826	1
Item25	-12.599567	-0.266091	-3.936308	0.751762	10.405225	1
Item26	-11.365093	-1.919706	-0.458052	1.861843	9.521104	1
Item27	-11.027619	-2.944884	-2.792962	4.144322	7.958556	1
Item28	-11.795160	-6.769646	0.908383	1.005066	11.240333	1
Item29	-13.629933	0.674184	-3.386853	-0.095859	10.490432	1
Item30	-7.823298	-5.452589	-2.336894	1.919889	9.421125	1
Item31	6.360118	6.794549	4.168188	-4.492538	-12.297555	0
Item32	8.774682	0.492721	1.587909	-5.486587	-12.361278	0
Item33	7.768181	3.989776	3.289377	-0.895444	-13.067171	0
Item34	8.581133	2.922361	3.952544	-4.450362	-6.787133	0
Item35	6.176519	4.526292	-2.771599	-3.477187	-7.316202	0
Item36	11.539781	-0.892880	2.868221	-1.456557	-11.008881	0
Item37	11.743034	3.527726	-0.635792	-2.067965	-7.151524	0
Item38	10.527299	1.460768	0.862300	-1.967742	-8.819727	0
Item39	8.148808	5.157964	-0.916135	-1.551958	-9.467513	0
Item40	9.241432	1.483108	-0.981933	1.046571	-8.504166	0
Item41	9.444197	4.963927	1.127201	-0.523484	-9.102817	0
Item42	11.545396	4.604968	4.818171	-5.046815	-13.494675	0
Item43	11.890988	1.220710	-2.069796	-2.942747	-8.996673	0
Item44	11.810480	2.031465	2.987976	-5.699606	-10.026246	0
Item45	10.806543	5.275155	4.969420	-2.792596	-11.345561	0
Item46	10.261177	3.586077	3.340220	-3.339244	-7.795038	0
Item47	8.407544	2.887997	3.104312	-3.734519	-9.758477	0
Item48	7.317484	5.553850	1.618000	-2.525315	-13.613147	0
Item49	10.654500	2.579577	1.922452	-3.765160	-10.414136	0
Item50	6.940641	3.525834	-0.660756	-4.105869	-10.064455	0

Parameters:

Input	
Data	File with the data in TABLE format. File should contain table data: lines for observations and columns for variables; columns should be separated by tabulation or user-defines sybol (; , etc); no missing data allowed.
List of variables 1	List of variables for which calculate variances, namely column indices. If ALL specified, program use all variables for analysis. Examples of input: 1;2;3-7;12; 1-12; ALL If 'Observation name' parameter is set on, variable list should not contain 1.
List of variables 2	List of variables for which calculate variances, namely column indices. If ALL specified, program use all variables for analysis. Examples of input: 1;2;3-7;12; 1-12; ALL If 'Observation name' parameter is set on, variable list should not contain 1.
Output	

Result	Name of output file
Options	
Field separation	Symbol for separation variables in line; by default tabulation and space.
Commentary line symbol	Commentary line symbol (if line starts from CommentSymbol, then this line is ignored)
Flip file before processing	Flip file before processing
Take Observation names from 1st column in table	Take Observation names from 1st column in table or Generate Observation names (Observation1,Observation2)

Variances

The program calculates variances of the values in columns of data in table format.

This program use statistical functions from "R" free software environment for statistical computing and graphics (http://www.r-project.org).

This program requires the R-package to be installed on your computer.

Program is provided with viewer.

Input file should contain table of numerical data: lines for observations (cases) columns should be separated by tabulation or user-defines symbol (; , etc); for example, if comma (,) separator is used, the file format is the same as the CSV (comma separated values) format. No missing data allowed.

Example of output data:

```
Variable	Variance
Feat1 101.8692
Feat2 14.1908
Feat3 6.0327
Feat4 10.8458
Feat5 102.6015
```

First line provides data description, separated by tabulation (Variable – names for selected variables; Variance – variances for variables). Next lines are the list variances for variables.

Example of input data file format:

ItemName	Feat1	Feat2	Feat3	Feat4	Feat5	ClassVar
Item1	-11.761101	-5.295846	-2.491684	4.151158	9.777093	1
Item2	-11.425886	-6.753716	0.136692	5.161748	13.618702	1
Item3	-7.069796	0.545457	0.097140	0.678579	10.302988	1
Item4	-13.480880	-3.867702	0.119297	2.333842	10.992096	1
Item5	-9.707938	-2.597949	-2.329997	2.928526	8.441053	1
Item6	-10.013794	-2.165258	-3.169195	2.625904	10.611103	1
Item7	-9.057161	-4.766594	1.691733	1.655782	7.046236	1
Item8	-8.562761	-1.272652	-3.990204	2.286294	12.768212	1
Item9	-12.724631	-4.710623	-2.114719	2.812189	6.434645	1
Item10	-9.593738	-5.478652	-1.799524	4.306497	9.514756	1
Item11	-7.699759	-1.546648	-0.423322	4.889767	9.228675	1
Item12	-13.158116	-2.891354	0.595935	2.264199	12.004761	1
Item13	-10.509598	-3.414075	-1.962310	1.263863	10.199896	1
Item14	-6.547624	-3.594928	-2.117222	5.168950	10.838221	1
Item15	-12.375988	-3.130436	-2.169164	1.537614	11.112888	1
Item16	-12.953032	-2.805048	0.085116	3.303354	7.405194	1
Item17	-11.370708	-2.848384	-0.848201	3.885525	10.569231	1
Item18	-13.117222	-7.025575	1.406507	7.069338	12.230415	1

Item19	-11.573168	0.288003	-2.826167	4.397137	10.851711	1
Item20	-7.993835	-1.204352	-1.924345	0.829829	10.314768	1
Item21	-9.225135	-2.512925	-1.608051	1.420301	9.766411	1
Item22	-8.402783	-0.890500	3.189703	3.754479	7.481063	1
Item23	-9.888180	-3.345775	1.965667	2.906369	11.488815	1
Item24	-11.686270	-5.389477	2.556932	1.661153	9.717826	1
Item25	-12.599567	-0.266091	-3.936308	0.751762	10.405225	1
Item26	-11.365093	-1.919706	-0.458052	1.861843	9.521104	1
Item27	-11.027619	-2.944884	-2.792962	4.144322	7.958556	1
Item28	-11.795160	-6.769646	0.908383	1.005066	11.240333	1
Item29	-13.629933	0.674184	-3.386853	-0.095859	10.490432	1
Item30	-7.823298	-5.452589	-2.336894	1.919889	9.421125	1
Item31	6.360118	6.794549	4.168188	-4.492538	-12.297555	0
Item32	8.774682	0.492721	1.587909	-5.486587	-12.361278	0
Item33	7.768181	3.989776	3.289377	-0.895444	-13.067171	0
Item34	8.581133	2.922361	3.952544	-4.450362	-6.787133	0
Item35	6.176519	4.526292	-2.771599	-3.477187	-7.316202	0
Item36	11.539781	-0.892880	2.868221	-1.456557	-11.008881	0
Item37	11.743034	3.527726	-0.635792	-2.067965	-7.151524	0
Item38	10.527299	1.460768	0.862300	-1.967742	-8.819727	0
Item39	8.148808	5.157964	-0.916135	-1.551958	-9.467513	0
Item40	9.241432	1.483108	-0.981933	1.046571	-8.504166	0
Item41	9.444197	4.963927	1.127201	-0.523484	-9.102817	0
Item42	11.545396	4.604968	4.818171	-5.046815	-13.494675	0
Item43	11.890988	1.220710	-2.069796	-2.942747	-8.996673	0
Item44	11.810480	2.031465	2.987976	-5.699606	-10.026246	0
Item45	10.806543	5.275155	4.969420	-2.792596	-11.345561	0
Item46	10.261177	3.586077	3.340220	-3.339244	-7.795038	0
Item47	8.407544	2.887997	3.104312	-3.734519	-9.758477	0
Item48	7.317484	5.553850	1.618000	-2.525315	-13.613147	0
Item49	10.654500	2.579577	1.922452	-3.765160	-10.414136	0
Item50	6.940641	3.525834	-0.660756	-4.105869	-10.064455	0

Parameters:

Input	
Data	File with the data in TABLE format. File should contain table data: lines for observations and columns for variables; columns should be separated by tabulation or user-defines sybol (; , etc); no missing data allowed.
List of variables	List of variables for which calculate variances, namely column indices. If ALL specified, program use all variables for analysis. Examples of input: 1;2;3-7;12; 1-12; ALL If 'Observation name' parameter is set on, variable list should not contain 1.
Output	
Result	Name of output file
Significant digits	Specifies the minimum number of significant digits to be printed in values.
XML data	Name of the file for graphical output.
Title	User-specified title of the graph plot.

Author	User-specified name of the graph author.
Comment	User-specified graph additional commentary line.
X axis name	User-specified graph X axis name.
Y axis name	User-specified graph Y axis name.
Options	
Field separation	Symbol for separation variables in line; by default tabulation and space.
Commentary line symbol	Commentary line symbol (if line starts from CommentSymbol, then this line is ignored)
Flip file before processing	Flip file before processing
Take Observation names from 1st column in table	Take Observation names from 1st column in table or Generate Observation names (Observation1,Observation2)

NN-Clust

Nearest Neighbor clustering

Perceptron

Perception Learning algorithm